



GLAST

The Gamma Ray Large Area Space Telescope

GLAST LAT ACD ELECTRONICS SUBSYSTEM TECHNICAL DOCUMENT	Document # ACD-PROC-000148	Date Effective 10-17-2003
	Prepared by: J. L. Odom and D. A. Sheppard	Supersedes None
Document Title GLAST LAT ACD FREE Short Functional Test		

**Gamma-ray Large Area Space Telescope (GLAST)
Large Area Telescope (LAT)
Anti-Coincidence Detector (ACD)**

FREE (Front End & Event Electronics)



Short Functional Test

Rev (–) October 17, 2003

Document Approval

Prepared by:

James Odom Date
ACD Electronics

Prepared by:

Dave Sheppard Date
ACD Electronics

Approved by:

Glenn Unger Date
ACD Electronics System Lead

Approved by:

George Shiblee Date
ACD Systems Engineer

Approved by:

Tom Johnson Date
ACD Project Manager

Approved by:

Dave Thompson Date
ACD Subsystems Manager

Approved by:

Eileen Fowler Date
Quality Assurance

Approved by:

Bob Hartman Date
ACD Scientist

FREE Short Functional Test Table of Contents

1.0	Scope and Purpose of this Procedure	6
2.0	Description of the FREE Electronics Assembly	6
3.0	Preconditions to and Preparations for Starting this Test.....	7
3.1	FREE Assembly Identification.....	7
3.2	Test Equipment Utilized	8
3.3	Suggested Equipment Setup.....	9
3.4	Verification of the GSE Ground.....	9
4.0	Electrical Safe-to-Mate Verification of the FREE Assembly	9
4.1	Verification of Proper LVDS Terminations.....	10
4.2	Stray Voltage Test at the AEM Interface	11
5.0	Functional Test	14
5.1	GARC and GAFE Registers Initial Reset Test	14
5.2	FREE Power Measurement at the Nominal Power Supply Voltage	15
5.3	FREE Power Measurement at the Minimum Power Supply Voltage	16
5.4	FREE Power Measurement at the Maximum Power Supply Voltage.....	17
5.5	GARC/GAFE Register Read/Write Tests	17
5.6	Veto_Delay Register Test	17
5.7	HVBS Level Register Test.....	18
5.8	SAA Level Register Test	18
5.9	Hold Delay Register Test	18
5.10	Veto Width Register Test.....	19
5.11	HitMap Width Register Test	19
5.12	HitMap Deadtime Register Test.....	19
5.13	HitMap Delay Register Test.....	19
5.14	PHA En0 Register Test	20
5.15	Veto En0 Register Test	20
5.16	PHA En1 Register Test	20
5.17	Veto En1 Register Test	20
5.18	MaxPHA Register Test	21
5.19	GARC Mode Register Test	21
5.20	PHA Threshold Channel 00 Register Test	22
5.21	PHA Threshold Channel 01 Register Test	22
5.22	PHA Threshold Channel 02 Register Test	22
5.23	PHA Threshold Channel 03 Register Test	22
5.24	PHA Threshold Channel 04 Register Test	23
5.25	PHA Threshold Channel 05 Register Test	23
5.26	PHA Threshold Channel 06 Register Test	23
5.27	PHA Threshold Channel 07 Register Test	24
5.28	PHA Threshold Channel 08 Register Test	24
5.29	PHA Threshold Channel 09 Register Test	24
5.30	PHA Threshold Channel 10 Register Test	24
5.31	PHA Threshold Channel 11 Register Test	25
5.32	PHA Threshold Channel 12 Register Test	25
5.33	PHA Threshold Channel 13 Register Test	25

5.34	PHA Threshold Channel 14 Register Test	26
5.35	PHA Threshold Channel 15 Register Test	26
5.36	PHA Threshold Channel 16 Register Test	26
5.37	PHA Threshold Channel 17 Register Test	27
5.38	ADC TACQ Register Test	27
5.39	GAFE ASICs Mode Register Test	27
5.40	GAFE ASIC VETO_DAC Register Test	28
5.41	GAFE ASIC VETO_VERNIER Register Test	28
5.42	GAFE ASIC HLD_DAC Register Test	28
5.43	GAFE ASIC BIAS_DAC Register Test	29
5.44	GAFE ASIC TCI_DAC Register Test	29
5.45	Initial GAFE Logic Reset Test	29
5.46	GAFE ASIC Broadcast Command Functional Verification	30
5.47	Test of the GARC Parity Error Detection and Error Generation	30
5.48	Test of the Look-At-Me Circuitry	31
5.49	Maximum PHA Return Test	32
5.50	PHA Enable/Disable Test	33
5.51	PHA Threshold Verification Test	34
5.52	Test of the HVBS DAC Commands, DAC Buffer, and Differential Drivers	35
5.53	Test of the HVBS Triple Modular Redundancy Circuitry	36
5.54	Test of the SAA/HV Normal Modes	38
5.55	Test of the GARC LVDS Circuitry Driver Currents	39
5.56	Characterization of each of the 18 GAFE chips	40
5.57	TCI and GAFE ADC Checkout	43
5.58	Power Supply Rail Tests – 3.0V to 3.6V	43
Appendix 1: GARC Documentation		45
Appendix 2: GARC Configuration Command Format		45
Appendix 3: GARC Event Data Format		46
Appendix 4: GARC Configuration Data Readback Format		46
Appendix 5: MDM Connector Pin List		47
Appendix 6: Series II Circular Connector Pin List		47
Appendix 7: GARC Command Mnemonics and Functions		49
Appendix 8: Test Results Record		53

Document Change Log

Revision	Date	Change Description	Prepared by
Initial Draft	September 29, 2003	Initial Draft	Jim Odom and Dave Sheppard
In Review	October 2, 2003	Updates from Review	Jim Odom and Dave Sheppard
Final Release	October 17, 2003	Final Release	Jim Odom and Dave Sheppard

1.0 Scope and Purpose of this Procedure

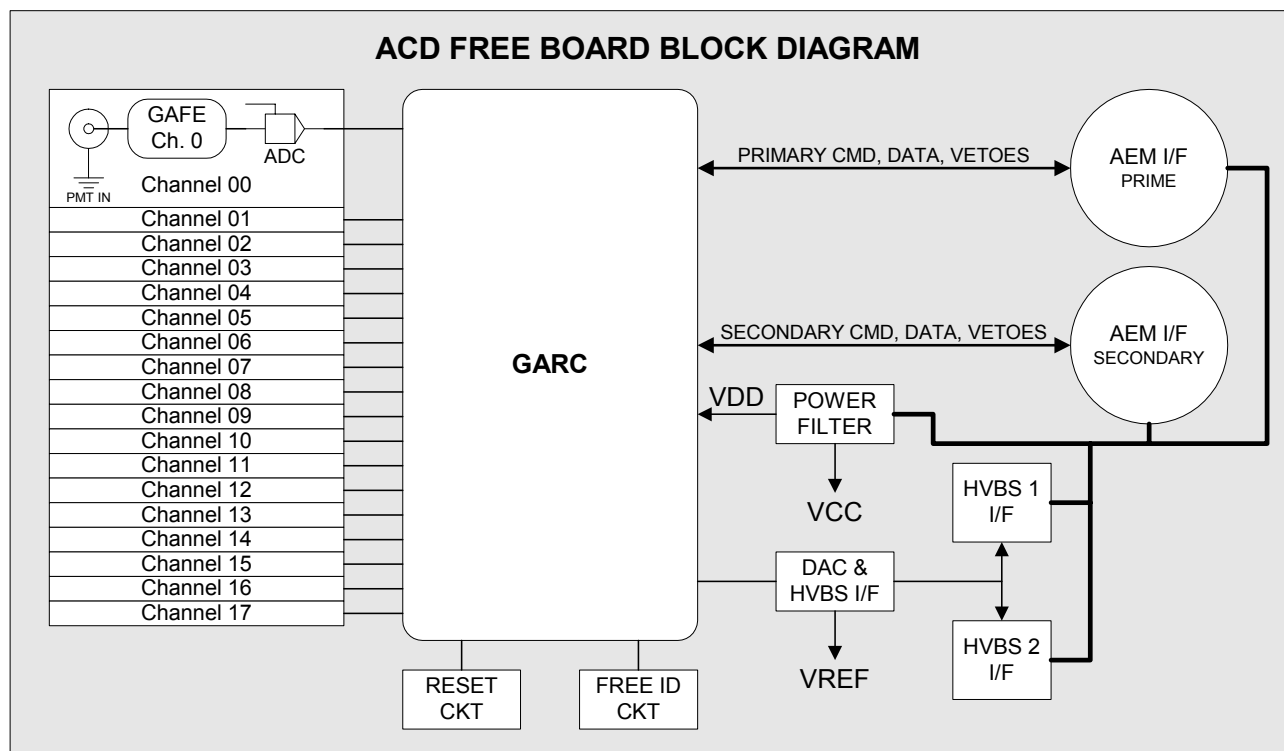
This document describes the instructions for the FREE assemblies Short Functional Test. It does not duplicate all the functions of component-level testing, but instead concentrates on assembly verification issues. This procedure provides the instructions necessary to perform a short functional test on the FREE assembly. This test is intended to be modified to provide other levels of functional testing as required during instrument and spacecraft level integrations.

The FREE assembly is designed to meet the electrical specifications of:

- (1) The ACD Level IV Electronics Requirements, LAT-SS-00352
- (2) The ACD-LAT Interface Control Document, LAT-SS-00-363

2.0 Description of the FREE Electronics Assembly

FREE is the acronym for the Gamma-Ray Large Area Space Telescope (GLAST) Anti-Coincidence Detector (ACD) Front End and Event Electronics processing board. The FREE provides the main electrical interface between the ACD and the LAT instrument electronics. The FREE circuitry provides command and data return functions for electronics boards associated with the ACD. There are 12 FREE boards in the flight system, with each FREE card supporting 18 phototube chains and 2 high voltage bias supplies (HVBS). The FREE circuitry utilizes a single +3.3V power supply and provides an interface to pass +28V power to the two HVBS. The FREE is a 10 layer flexible-rigid printed circuit board. It has 18 analog signal processing channels serviced by a single GARC readout controller. It also provides the interface to the HVBS and has two thermistors on board.



The design of the FREE circuitry was done utilizing Orcad for schematic capture, PSpice for simulation, and PADS for board layout. The current copy of the schematic package is posted in PDF format on the ACD website at: <http://lhea-glast.gsfc.nasa.gov/acd/electronics/#free>

3.0 Preconditions to and Preparations for Starting this Test

Prior to starting this test, a responsible Test Conductor shall be named. The Test Conductor is responsible for the safety of the hardware and the documentation of results, including anomalies, for the duration of the test. The Test Conductor, or a designated representative, shall be present for each testing sequence. All testing on flight hardware will require a signed Work Authorization Order. Each person working directly with the ACD flight hardware shall be NASA certified for electrostatic discharge control as per NASA-STD-8739.7. All measurement equipment used for verification tests on flight hardware shall have a valid calibration sticker. The Test Conductor shall have the authority to determine deviations from this procedure and shall redline this procedure as necessary. Mates and demates to flight connectors shall be recorded. Connectors shall not be mated nor demated unless the instrument electronics is powered off. Connector savers shall be utilized wherever practical to minimize flight connector mate/demates.

Notify QA 24 hours prior to the start of this test. QA will verify test equipment set up and calibration, review documentation and decide if their presence is required during the testing. Indicate on the test record whether QA was present for the testing.

All measurements and test results will be recorded in the "Test Results Record" Appendix 11. A copy of the entire this entire document is not required each time the test is performed. The appendix will serve as the official record each time this test is performed.

This test will be run at various times during the build-up and Integration of the FREE boards. This test can be run only after the FREE CPT ACD-PROC-000051 has been run in its entirety during the first turn-on and check-out. If the Test Conductor deems that a test should be omitted due to a test point not being accessible or poses a risk to the board the Test Conductor may omit the test by making a note on the Test Record.

3.1 FREE Assembly Identification

The test conductor for this test is: _____

Date of Test: _____

The serial number of the FREE card assembly is: _____

The identification/Serial Number listed on the GARC ASIC is: _____

The identification/Serial Numbers listed on the GAFE ASICs are:

GAFE Channel	ASIC Serial Number
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

12	
13	
14	
15	
16	
17	

Note that for flight assemblies, the performance of this test must be listed on the appropriate Work Authorization Order (WOA).

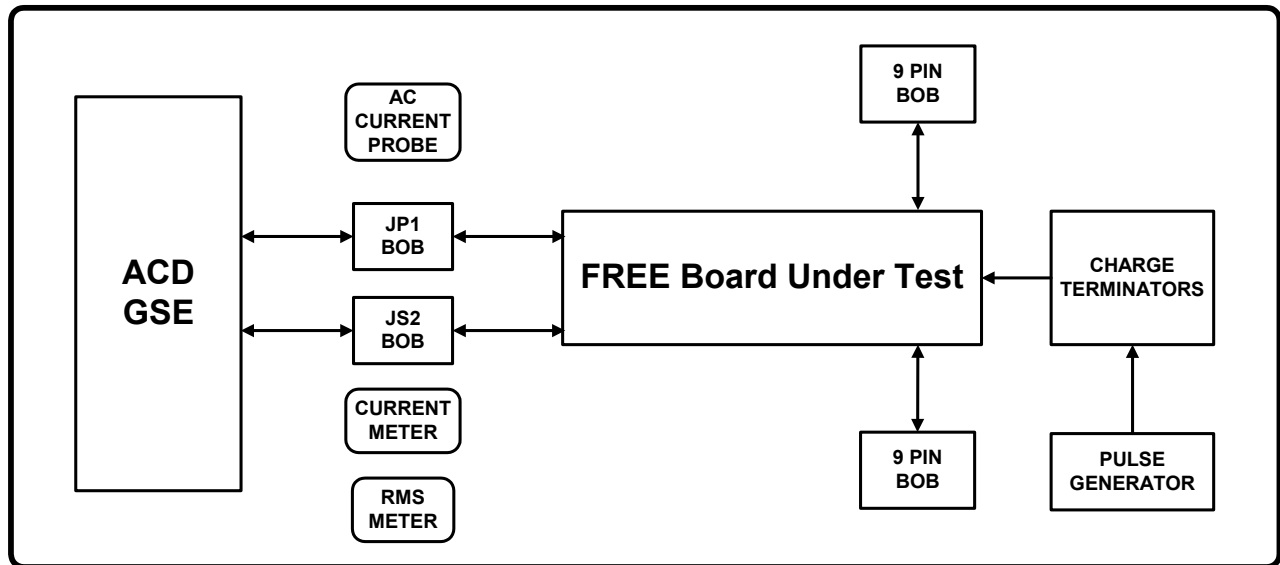
3.2 Test Equipment Utilized

Prior to starting this test, the Test Conductor shall record the test and measurement hardware used in the performance of this test. Note that all multimeters used in testing of the FREE circuit card assembly shall be of low current output design, such as the Fluke 70 series, HP3400 series or similar. High current output multimeters are not compatible with the FREE circuitry.

Instrument Type	Manufacturer & Model Number	NASA ID Number	Calibration Due Date
Power Supply for HVBS +28V			
Power Supply for FREE & GSE +3.3V & +5.0V			
Multimeter 1			
Multimeter 2			
Tail Pulse Generator			
Oscilloscope			
Pulse Generator			
Charge Terminator Box			
GSE Used			

3.3 Suggested Equipment Setup

Prior to starting the test, it is recommended that the laboratory bench be set up as follows:



3.4 Verification of the GSE Ground

Prior to the connection of FREE assembly hardware to other electronics, it shall be verified that all power supplies, signal generators, VME racks, and any other test and measurement equipment shall be connected to the same AC ground. The simplest way to do this is to connect all AC-powered equipment to the same power strip. In cases where this is not practical (e.g., possibly a thermal-vacuum test), greater care must be taken to ensure there are no floating grounds since this would represent a hazard to the FREE assembly.

4.0 Electrical Safe-to-Mate Verification of the FREE Assembly

The intent of this section is to ensure that the FREE assembly is both electrically and mechanically safe to mate to the AEM or AEM GSE to be used for testing.

Break Out Boxes (BOB) and a calibrated multimeter will be required for this test.

This section is to be performed prior to doing electrical testing on the FREE assembly. If this verification has been performed successfully, the Test Conductor may opt to omit this portion of the test if both the FREE assembly and the AEM interface have been checked during prior tests.

1. Verify that connector savers have been installed on FREE connectors JP1, JS2, JHV1, and JHV2.
2. Verify that break out boxes (BOB) are available for these four connectors. These BOB should have been pin-for-pin checked for continuity and isolation prior to the start of this procedure.
3. The test conductor shall visually inspect each of the connector halves on the FREE assembly, BOB, harnessing, and AEM interface to ensure there are no bent pins, debris, or other physical damage that would impede safe connector mating.
4. All shorting pins should be removed from the BOBs.
5. Verify that the power supplies are OFF.
6. Mate the four BOBs to connectors JP1, JS2, JHV1 and JHV2 on the FREE assembly. Mate the other side of the BOBs to the AEM interface connectors.

4.1 Verification of Proper LVDS Terminations

The ACD interface requires proper termination of the LVDS drivers. The specification is $100\ \Omega \pm 5\%$. This test is performed with the ACD power off. A multimeter is used to measure the impedance across the following signal pairs on the GSE side of the BOB and the result is recorded

This test may be omitted at the discretion of the Test Conductor.

Measurement No.	Signal Pair	Pin 1	Pin 2	Expected Resistance	Verified
1	ACD_NVETO_16_A	JP1 - 17	JP1 - 18	$95\ \Omega < R < 105\ \Omega$	
2	ACD_NVETO_17_A	JP1 - 19	JP1 - 20	$95\ \Omega < R < 105\ \Omega$	
3	ACD_CNO_A	JP1 - 21	JP1 - 22	$95\ \Omega < R < 105\ \Omega$	
4	ACD_NVETO_15_A	JP1 - 40	JP1 - 41	$95\ \Omega < R < 105\ \Omega$	
5	ACD_NVETO_14_A	JP1 - 42	JP1 - 43	$95\ \Omega < R < 105\ \Omega$	
6	ACD_NVETO_13_A	JP1 - 44	JP1 - 45	$95\ \Omega < R < 105\ \Omega$	
7	ACD_NVETO_12_A	JP1 - 46	JP1 - 47	$95\ \Omega < R < 105\ \Omega$	
8	ACD_NVETO_11_A	JP1 - 48	JP1 - 49	$95\ \Omega < R < 105\ \Omega$	
9	ACD_NVETO_10_A	JP1 - 50	JP1 - 51	$95\ \Omega < R < 105\ \Omega$	
10	ACD_NVETO_09_A	JP1 - 52	JP1 - 53	$95\ \Omega < R < 105\ \Omega$	
11	ACD_NVETO_08_A	JP1 - 54	JP1 - 55	$95\ \Omega < R < 105\ \Omega$	
12	ACD_NVETO_07_A	JP1 - 56	JP1 - 57	$95\ \Omega < R < 105\ \Omega$	
13	ACD_NVETO_06_A	JP1 - 58	JP1 - 59	$95\ \Omega < R < 105\ \Omega$	
14	ACD_NVETO_05_A	JP1 - 60	JP1 - 61	$95\ \Omega < R < 105\ \Omega$	
15	ACD_NVETO_04_A	JP1 - 62	JP1 - 63	$95\ \Omega < R < 105\ \Omega$	
16	ACD_NVETO_03_A	JP1 - 64	JP1 - 65	$95\ \Omega < R < 105\ \Omega$	
17	ACD_NVETO_02_A	JP1 - 66	JP1 - 67	$95\ \Omega < R < 105\ \Omega$	
18	ACD_NVETO_01_A	JP1 - 68	JP1 - 69	$95\ \Omega < R < 105\ \Omega$	
19	ACD_NVETO_00_A	JP1 - 70	JP1 - 71	$95\ \Omega < R < 105\ \Omega$	
20	ACD_NSDATA_A	JP1 - 72	JP1 - 73	$95\ \Omega < R < 105\ \Omega$	
21	ACD_NRST_A	JP1 - 74	JP1 - 75	$R > 100\ \text{k}\Omega$	
22	ACD_NSCMD_A	JP1 - 76	JP1 - 77	$R > 100\ \text{k}\Omega$	
23	ACD_CLK_A	JP1 - 78	JP1 - 79	$R > 100\ \text{k}\Omega$	
24	ACD_NVETO_16_B	JS2 - 17	JS2 - 18	$95\ \Omega < R < 105\ \Omega$	
25	ACD_NVETO_17_B	JS2 - 19	JS2 - 20	$95\ \Omega < R < 105\ \Omega$	
26	ACD_CNO_B	JS2 - 21	JS2 - 22	$95\ \Omega < R < 105\ \Omega$	
27	ACD_NVETO_15_B	JS2 - 40	JS2 - 41	$95\ \Omega < R < 105\ \Omega$	
28	ACD_NVETO_14_B	JS2 - 42	JS2 - 43	$95\ \Omega < R < 105\ \Omega$	
29	ACD_NVETO_13_B	JS2 - 44	JS2 - 45	$95\ \Omega < R < 105\ \Omega$	
30	ACD_NVETO_12_B	JS2 - 46	JS2 - 47	$95\ \Omega < R < 105\ \Omega$	
31	ACD_NVETO_11_B	JS2 - 48	JS2 - 49	$95\ \Omega < R < 105\ \Omega$	
32	ACD_NVETO_10_B	JS2 - 50	JS2 - 51	$95\ \Omega < R < 105\ \Omega$	
33	ACD_NVETO_09_B	JS2 - 52	JS2 - 53	$95\ \Omega < R < 105\ \Omega$	
34	ACD_NVETO_08_B	JS2 - 54	JS2 - 55	$95\ \Omega < R < 105\ \Omega$	
35	ACD_NVETO_07_B	JS2 - 56	JS2 - 57	$95\ \Omega < R < 105\ \Omega$	
36	ACD_NVETO_06_B	JS2 - 58	JS2 - 59	$95\ \Omega < R < 105\ \Omega$	
37	ACD_NVETO_05_B	JS2 - 60	JS2 - 61	$95\ \Omega < R < 105\ \Omega$	
38	ACD_NVETO_04_B	JS2 - 62	JS2 - 63	$95\ \Omega < R < 105\ \Omega$	
39	ACD_NVETO_03_B	JS2 - 64	JS2 - 65	$95\ \Omega < R < 105\ \Omega$	
40	ACD_NVETO_02_B	JS2 - 66	JS2 - 67	$95\ \Omega < R < 105\ \Omega$	
41	ACD_NVETO_01_B	JS2 - 68	JS2 - 69	$95\ \Omega < R < 105\ \Omega$	

42	ACD_NVETO_00_B	JS2 - 70	JS2 - 71	$95\ \Omega < R < 105\ \Omega$	
43	ACD_NSDATA_B	JS2 - 72	JS2 - 73	$95\ \Omega < R < 105\ \Omega$	
44	ACD_NIRST_B	JS2 - 74	JS2 - 75	$R > 100\ \text{k}\Omega$	
45	ACD_NSCMD_B	JS2 - 76	JS2 - 77	$R > 100\ \text{k}\Omega$	
46	ACD_CLK_B	JS2 - 78	JS2 - 79	$R > 100\ \text{k}\Omega$	

At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

4.2 Stray Voltage Test at the AEM Interface

With the shorting plugs removed from the break out boxes, turn the ACD power on and perform the following voltage measurements on the AEM interface side of the break out box. The (-) input to the voltmeter may be referenced to pin 30, the +3.3V return.

If this verification has been performed successfully, the Test Conductor may opt to omit this portion of the test if the AEM interface has been checked.

Meas. No.	AEM Interface Pin	Signal Name	Expected Voltage	Measured Voltage
1	JP1 - 1	ACD_VDD_0A	+3.3V	
2	JP1 - 3	ACD_VDD_1A	+3.3V	
3	JP1 - 4	ACD_VDD_2A	+3.3V	
4	JP1 - 5	ACD_28V_0A	+28V	
5	JP1 - 7	ACD_28V_1A	+28V	
6	JP1 - 17	ACD_NVETO_16AP	$0 < V < 3.3$	
7	JP1 - 18	ACD_NVETO_16AM	$0 < V < 3.3$	
8	JP1 - 19	ACD_NVETO_17AP	$0 < V < 3.3$	
9	JP1 - 20	ACD_NVETO_17AM	$0 < V < 3.3$	
10	JP1 - 21	ACD_NCNO_AP	$0 < V < 3.3$	
11	JP1 - 22	ACD_NCNO_AM	$0 < V < 3.3$	
12	JP1 - 23	ACD_HV_AP	$0 < V < 3.3$	
13	JP1 - 24	ACD_HV_AM	$0 < V < 3.3$	
14	JP1 - 25	ACD_TEMP_AP	TBD	
15	JP1 - 26	ACD_TEMP_AM	TBD	
16	JP1 - 30	ACD_GND_0A	0	
17	JP1 - 31	ACD_GND_1A	0	
18	JP1 - 32	ACD_GND_2A	0	
19	JP1 - 33	ACD_28V_RTN_0A	0	
20	JP1 - 34	ACD_28V_RTN_1A	0	
21	JP1 - 40	ACD_NVETO_15AM	$V < 200\text{mV}$	
22	JP1 - 41	ACD_NVETO_15AP	$V < 200\text{mV}$	
23	JP1 - 42	ACD_NVETO_14AM	$V < 200\text{mV}$	
24	JP1 - 43	ACD_NVETO_14AP	$V < 200\text{mV}$	
25	JP1 - 44	ACD_NVETO_13AM	$V < 200\text{mV}$	
26	JP1 - 45	ACD_NVETO_13AP	$V < 200\text{mV}$	
27	JP1 - 46	ACD_NVETO_12AM	$V < 200\text{mV}$	
28	JP1 - 47	ACD_NVETO_12AP	$V < 200\text{mV}$	
29	JP1 - 48	ACD_NVETO_11AM	$V < 200\text{mV}$	
31	JP1 - 50	ACD_NVETO_10AM	$V < 200\text{mV}$	

32	JP1 - 51	ACD_NVETO_10AP	V < 200mV	
33	JP1 - 52	ACD_NVETO_09AM	V < 200mV	
34	JP1 - 53	ACD_NVETO_09AP	V < 200mV	
35	JP1 - 54	ACD_NVETO_08AM	V < 200mV	
36	JP1 - 55	ACD_NVETO_08AP	V < 200mV	
37	JP1 - 56	ACD_NVETO_07AM	V < 200mV	
38	JP1 - 57	ACD_NVETO_07AP	V < 200mV	
39	JP1 - 58	ACD_NVETO_06AM	V < 200mV	
40	JP1 - 59	ACD_NVETO_06AP	V < 200mV	
41	JP1 - 60	ACD_NVETO_05AM	V < 200mV	
42	JP1 - 61	ACD_NVETO_05AP	V < 200mV	
43	JP1 - 62	ACD_NVETO_04AM	V < 200mV	
44	JP1 - 63	ACD_NVETO_04AP	V < 200mV	
45	JP1 - 64	ACD_NVETO_03AM	V < 200mV	
46	JP1 - 65	ACD_NVETO_03AP	V < 200mV	
47	JP1 - 66	ACD_NVETO_02AM	V < 200mV	
48	JP1 - 67	ACD_NVETO_02AP	V < 200mV	
49	JP1 - 68	ACD_NVETO_01AM	V < 200mV	
50	JP1 - 69	ACD_NVETO_01AP	V < 200mV	
51	JP1 - 70	ACD_NVETO_00AM	V < 200mV	
52	JP1 - 71	ACD_NVETO_00AP	V < 200mV	
53	JP1 - 72	ACD_NSDATA_AM	V < 200mV	
54	JP1 - 73	ACD_NSDATA_AP	V < 200mV	
55	JP1 - 74	ACD_NRST_AM	V < 200mV	
56	JP1 - 75	ACD_NRST_AP	V > 2.3V	
57	JP1 - 76	ACD_NSCMD_AM	V < 200mV	
58	JP1 - 77	ACD_NSCMD_AP	V > 2.3V	
59	JP1 - 78	ACD_CLK_AM	1.1V < V < 1.5V	
60	JP1 - 79	ACD_CLK_AP	1.1V < V < 1.5V	
61	JS2 - 1	ACD_VDD_0B	+3.3V	
62	JS2 - 3	ACD_VDD_1B	+3.3V	
63	JS2 - 4	ACD_VDD_2B	+3.3V	
64	JS2 - 5	ACD_28V_0B	+28V	
65	JS2 - 7	ACD_28V_1B	+28V	
66	JS2 - 17	ACD_NVETO_16BP	V < 200mV	
67	JS2 - 18	ACD_NVETO_16BM	V < 200mV	
68	JS2 - 19	ACD_NVETO_17BP	V < 200mV	
69	JS2 - 20	ACD_NVETO_17BM	V < 200mV	
70	JS2 - 21	ACD_NCNO_BP	V < 200mV	
71	JS2 - 22	ACD_NCNO_BM	V < 200mV	
72	JS2 - 23	ACD_HV_BP	V < 200mV	
73	JS2 - 24	ACD_HV_BM	V < 200mV	
74	JS2 - 25	ACD_TEMP_BP	TBD	
75	JS2 - 26	ACD_TEMP_BM	TBD	
76	JS2 - 30	ACD_GND_0B	0	
77	JS2 - 31	ACD_GND_1B	0	
78	JS2 - 32	ACD_GND_2B	0	
79	JS2 - 33	ACD_28V_RTN_0B	0	
80	JS2 - 34	ACD_28V_RTN_1B	0	
81	JS2 - 40	ACD_NVETO_15BM	V < 200mV	
82	JS2 - 41	ACD_NVETO_15BP	V < 200mV	

83	JS2 - 42	ACD_NVETO_14BM	V < 200mV	
84	JS2 - 43	ACD_NVETO_14BP	V < 200mV	
85	JS2 - 44	ACD_NVETO_13BM	V < 200mV	
86	JS2 - 45	ACD_NVETO_13BP	V < 200mV	
87	JS2 - 46	ACD_NVETO_12BM	V < 200mV	
88	JS2 - 47	ACD_NVETO_12BP	V < 200mV	
89	JS2 - 48	ACD_NVETO_11BM	V < 200mV	
90	JS2 - 49	ACD_NVETO_11BP	V < 200mV	
91	JS2 - 50	ACD_NVETO_10BM	V < 200mV	
92	JS2 - 51	ACD_NVETO_10BP	V < 200mV	
93	JS2 - 52	ACD_NVETO_09BM	V < 200mV	
94	JS2 - 53	ACD_NVETO_09BP	V < 200mV	
95	JS2 - 54	ACD_NVETO_08BM	V < 200mV	
96	JS2 - 55	ACD_NVETO_08BP	V < 200mV	
97	JS2 - 56	ACD_NVETO_07BM	V < 200mV	
98	JS2 - 57	ACD_NVETO_07BP	V < 200mV	
99	JS2 - 58	ACD_NVETO_06BM	V < 200mV	
100	JS2 - 59	ACD_NVETO_06BP	V < 200mV	
101	JS2 - 60	ACD_NVETO_05BM	V < 200mV	
102	JS2 - 61	ACD_NVETO_05BP	V < 200mV	
103	JS2 - 62	ACD_NVETO_04BM	V < 200mV	
104	JS2 - 63	ACD_NVETO_04BP	V < 200mV	
105	JS2 - 64	ACD_NVETO_03BM	V < 200mV	
106	JS2 - 65	ACD_NVETO_03BP	V < 200mV	
107	JS2 - 66	ACD_NVETO_02BM	V < 200mV	
108	JS2 - 67	ACD_NVETO_02BP	V < 200mV	
109	JS2 - 68	ACD_NVETO_01BM	V < 200mV	
110	JS2 - 69	ACD_NVETO_01BP	V < 200mV	
111	JS2 - 70	ACD_NVETO_00BM	V < 200mV	
112	JS2 - 71	ACD_NVETO_00BP	V < 200mV	
113	JS2 - 72	ACD_NSDATA_BM	V < 200mV	
114	JS2 - 73	ACD_NSDATA_BP	V < 200mV	
115	JS2 - 74	ACD_NRST_BM	V < 200mV	
116	JS2 - 75	ACD_NRST_BP	V > 2.3V	
117	JS2 - 76	ACD_NSCMD_BM	V < 200mV	
118	JS2 - 77	ACD_NSCMD_BP	V > 2.3V	
119	JS2 - 78	ACD_CLK_BM	1.1V < V < 1.5V	
120	JS2 - 79	ACD_CLK_BP	1.1V < V < 1.5V	

At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

5.0 Functional Test

This section of the test will verify the general operation of the FREE circuit assembly. The initial power levels, the command and data interface, the DAC commanding, the GAFE test charge injection, the pulse height analysis (PHA) circuitry, the HVBS Interface, and the GAFE characterizations will be checked. The pin connections for the two 79 pin circular connectors JP1 and JS2 are listed in Appendix 6. The pin connections for the two 9 pin MDM connectors to the High Voltage Bias Supplies are listed in Appendix 5.

5.1 GARC and GAFE Registers Initial Reset Test

This section will verify that GARC registers have been properly initialized during a reset command. The following test sequence of commands will perform this verification. This test will also verify that the GARC to AEM command and data return link is functional. At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

Turn on the power supplies and measure the currents –

+3.3V Current - _____

+28V Current - _____

1. Verify the external GARC Power On Reset pulse. If using the LabView GSE, send the Look at me and the Read All Values command.
2. Send the Veto_Delay_Rd command. The data field in the return data stream should be 5.
3. Send the HVBS_Level_Rd command. The data field in the return data stream should be 0.
4. Send the SAA_Level_Rd command. The data field in the return data stream should be 0.
5. Send the Hold_Delay_Rd command. The data field in the return data stream should be 28.
6. Send the Veto_Width_Rd command. The data field in the return data stream should be 2.
7. Send the HitMap_Width_Rd command. The data field in the return data stream should be 7.
8. Send the HitMap_Deadtime_Rd command. The data field in the return data stream should be 3.
9. Send the HitMap_Delay_Rd command. The data field in the return data stream should be 16.
10. Send the PHA_En0_Rd command. The data field in the return data stream should be 65535.
11. Send the PHA_En1_Rd command. The data field in the return data stream should be 3.
12. Send the Veto_En0_Rd command. The data field in the return data stream should be 65535.
13. Send the Veto_En1_Rd command. The data field in the return data stream should be 3.
14. Send the Max_PHA_Rd command. The data field in the return data stream should be 4.
15. Send the GARC_Mode_Rd command. The data field in the return data stream should be 768.
16. Send the GARC_Status command. The data field in the return data stream should be 24.
17. Send the GARC_Cmd_Reg command. The data field in the return data stream should be 0.
18. Send the GARC_Cmd_Rejects command. The data field in the return data stream should be 0.

19. Send the GARC_Version command. The data field in the return data stream should be 3 for GARC V3 ASICs.
20. Send the PHA_Thresh00_Rd command. The data field in the return data stream should be 1114.
21. Send the PHA_Thresh01_Rd command. The data field in the return data stream should be 1114.
22. Send the PHA_Thresh02_Rd command. The data field in the return data stream should be 1114.
23. Send the PHA_Thresh03_Rd command. The data field in the return data stream should be 1114.
24. Send the PHA_Thresh04_Rd command. The data field in the return data stream should be 1114.
25. Send the PHA_Thresh05_Rd command. The data field in the return data stream should be 1114.
26. Send the PHA_Thresh06_Rd command. The data field in the return data stream should be 1114.
27. Send the PHA_Thresh07_Rd command. The data field in the return data stream should be 1114.
28. Send the PHA_Thresh08_Rd command. The data field in the return data stream should be 1114.
29. Send the PHA_Thresh09_Rd command. The data field in the return data stream should be 1114.
30. Send the PHA_Thresh10_Rd command. The data field in the return data stream should be 1114.
31. Send the PHA_Thresh11_Rd command. The data field in the return data stream should be 1114.
32. Send the PHA_Thresh12_Rd command. The data field in the return data stream should be 1114.
33. Send the PHA_Thresh13_Rd command. The data field in the return data stream should be 1114.
34. Send the PHA_Thresh14_Rd command. The data field in the return data stream should be 1114.
35. Send the PHA_Thresh15_Rd command. The data field in the return data stream should be 1114.
36. Send the PHA_Thresh16_Rd command. The data field in the return data stream should be 1114.
37. Send the PHA_Thresh17_Rd command. The data field in the return data stream should be 1114.
38. Send the ADC_TACQ_Rd command. The data field in the return data stream should be 0.
39. Send the Trigger_ZS command (this captures the FREE board ID). FREE board ID should be as recorded on the WOA.

Record FREE board ID - _____

40. Send the GARC_Reset command (Do Reset Pulse from the Labview Program).
41. Check the FREE board ID it should be 255 indicating that a reset had occurred..
42. Verify the status of the GARC registers as shown in steps 2 – 39 above.

If all steps above have verified correctly, the GARC reset function has been verified. A summary of the initial reset parameters for GARC and GAFE is detailed below.

5.2 FREE Power Measurement at the Nominal Power Supply Voltage

Verify that the GARC power supply is set to +3.30V. After initial power up, measure the +3.30V power supply current to the GARC in the following modes. Record these values in the table below.

1. Send the GARC_Mode_Wr command with a data argument of 768. Send the GARC_Mode_Rd command to verify. Both the primary and secondary LVDS VETO drivers should be enabled. Record the GARC current in the table below.

2. Send the GARC_Mode_Wr command with a data argument of 256. Send the GARC_Mode_Rd command to verify. Only the primary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
3. Send the GARC_Mode_Wr command with a data argument of 512. Send the GARC_Mode_Rd command to verify. Only the secondary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
4. Send the GARC_Mode_Wr command with a data argument of 0. Send the GARC_Mode_Rd command to verify. None of the LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

GARC Mode	GARC_Mode_Wr Data Argument	+3.3V Current Measured (mA)	+3.3V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		200 ± 10mA
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		150 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		150 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		95 ± 10mA

5.3 FREE Power Measurement at the Minimum Power Supply Voltage

Verify that the GARC power supply is set to +3.0V. After initial power up, measure the +3.0V power supply current to the GARC in the following modes. Record these values in the table below.

1. Send the GARC_Mode_Wr command with a data argument of 768. Send the GARC_Mode_Rd command to verify. Both the primary and secondary LVDS VETO drivers should be enabled. Record the GARC current in the table below.
2. Send the GARC_Mode_Wr command with a data argument of 256. Send the GARC_Mode_Rd command to verify. Only the primary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
3. Send the GARC_Mode_Wr command with a data argument of 512. Send the GARC_Mode_Rd command to verify. Only the secondary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
4. Send the GARC_Mode_Wr command with a data argument of 0. Send the GARC_Mode_Rd command to verify. None of the LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

GARC Mode	GARC_Mode_Wr Data Argument	+3.0V Current Measured (mA)	+3.0V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		170 ± 10mA
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		125 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		125 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		75 ± 10mA

5.4 FREE Power Measurement at the Maximum Power Supply Voltage

Verify that the GARC power supply is set to +3.6V. After initial power up, measure the +3.6V power supply current to the GARC in the following modes. Record these values in the table below.

1. Send the GARC_Mode_Wr command with a data argument of 768. Send the GARC_Mode_Rd command to verify. Both the primary and secondary LVDS VETO drivers should be enabled. Record the GARC current in the table below.
2. Send the GARC_Mode_Wr command with a data argument of 256. Send the GARC_Mode_Rd command to verify. Only the primary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
3. Send the GARC_Mode_Wr command with a data argument of 512. Send the GARC_Mode_Rd command to verify. Only the secondary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
4. Send the GARC_Mode_Wr command with a data argument of 0. Send the GARC_Mode_Rd command to verify. None of the LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

GARC Mode	GARC_Mode_Wr Data Argument	+3.6V Current Measured (mA)	+3.6V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		230 ± 10mA
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		175 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		175 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		115 ± 10mA

5. Reset the GARC power supply to +3.3V.
6. Send the GARC_Reset command to return the GARC to the initial power-on configuration.

5.5 GARC/GAFE Register Read/Write Tests

This section tests the proper functioning of each bit of the commandable registers in the GARC. The intent is to toggle each bit in a variety of patterns to ensure that all bits are addressable and that there is no stuck-at-fault condition. This can be accomplished using the LabView GSE software by doing the Register Test 3 times with all GAFE registers enabled.

After running the script proceed to step 5.45.

5.6 Veto Delay Register Test

1. Send the Veto_Delay_Wr command with a data argument of 5'h0 (0). Send the Veto_Delay_Rd command and read back this commanded data argument.
2. Send the Veto_Delay_Wr command with a data argument of 5'h15 (21). Send the Veto_Delay_Rd command and read back this commanded data argument.
3. Send the Veto_Delay_Wr command with a data argument of 5'h0A (10). Send the Veto_Delay_Rd command and read back this commanded data argument.

4. Send the Veto_Delay_Wr command with a data argument of 5'h1F (31). Send the Veto_Delay_Rd command and read back this commanded data argument.
5. Send the Veto_Delay_Wr command with a data argument of 5'h5 (5). Send the Veto_Delay_Rd command and read back this commanded data argument.

5.7 HVBS Level Register Test

1. Send the HVBS_Level_Wr command with a data argument of 12'h0 (0). Send the HVBS_Level_Rd command and read back this commanded data argument.
2. Send the HVBS_Level_Wr command with a data argument of 12'h555 (1365). Send the HVBS_Level_Rd command and read back this commanded data argument.
3. Send the HVBS_Level_Wr command with a data argument of 12'hAAA (2730). Send the HVBS_Level_Rd command and read back this commanded data argument.
4. Send the HVBS_Level_Wr command with a data argument of 12'hFFF (4095). Send the HVBS_Level_Rd command and read back this commanded data argument.
5. Send the HVBS_Level_Wr command with a data argument of 12'h0 (0). Send the HVBS_Level_Rd command and read back this commanded data argument.

5.8 SAA Level Register Test

1. Send the SAA_Level_Wr command with a data argument of 12'h0 (0). Send the SAA_Level_Rd command and read back this commanded data argument.
2. Send the SAA_Level_Wr command with a data argument of 12'h555 (1365). Send the SAA_Level_Rd command and read back this commanded data argument.
3. Send the SAA_Level_Wr command with a data argument of 12'hAAA (2730). Send the SAA_Level_Rd command and read back this commanded data argument.
4. Send the SAA_Level_Wr command with a data argument of 12'hFFF (4095). Send the SAA_Level_Rd command and read back this commanded data argument.
5. Send the SAA_Level_Wr command with a data argument of 12'h0 (0). Send the SAA_Level_Rd command and read back this commanded data argument.

5.9 Hold Delay Register Test

1. Send the Hold_Delay_Wr command with a data argument of 7'h0 (0). Send the Hold_Delay_Rd command and read back this commanded data argument.
2. Send the Hold_Delay_Wr command with a data argument of 7'h55 (85). Send the Hold_Delay_Rd command and read back this commanded data argument.
3. Send the Hold_Delay_Wr command with a data argument of 7'h2A (42). Send the Hold_Delay_Rd command and read back this commanded data argument.
4. Send the Hold_Delay_Wr command with a data argument of 7'h7F (127). Send the Hold_Delay_Rd command and read back this commanded data argument.
5. Send the Hold_Delay_Wr command with a data argument of 7'h1C (28). Send the Hold_Delay_Rd command and read back this commanded data argument.

5.10 Veto Width Register Test

1. Send the Veto_Width_Wr command with a data argument of 3'h0 (0). Send the Veto_Width_Rd command and read back this commanded data argument.
2. Send the Veto_Width_Wr command with a data argument of 3'h5 (5). Send the Veto_Width_Rd command and read back this commanded data argument.
3. Send the Veto_Width_Wr command with a data argument of 3'h7 (7). Send the Veto_Width_Rd command and read back this commanded data argument.
4. Send the Veto_Width_Wr command with a data argument of 3'h2 (2). Send the Veto_Width_Rd command and read back this commanded data argument.

5.11 HitMap Width Register Test

1. Send the HitMap_Width_Wr command with a data argument of 4'h0 (0). Send the HitMap_Width_Rd command and read back this commanded data argument.
2. Send the HitMap_Width_Wr command with a data argument of 4'h5 (5). Send the HitMap_Width_Rd command and read back this commanded data argument.
3. Send the HitMap_Width_Wr command with a data argument of 4'hA (10). Send the HitMap_Width_Rd command and read back this commanded data argument.
4. Send the HitMap_Width_Wr command with a data argument of 4'hF (15). Send the HitMap_Width_Rd command and read back this commanded data argument.
5. Send the HitMap_Width_Wr command with a data argument of 4'h7 (7). Send the HitMap_Width_Rd command and read back this commanded data argument.

5.12 HitMap Deadtime Register Test

1. Send the HitMap_Deadtime_Wr command with a data argument of 3'h0 (0). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
2. Send the HitMap_Deadtime_Wr command with a data argument of 3'h5 (5). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
3. Send the HitMap_Deadtime_Wr command with a data argument of 3'h2 (2). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
4. Send the HitMap_Deadtime_Wr command with a data argument of 3'h7 (7). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
5. Send the HitMap_Deadtime_Wr command with a data argument of 3'h3 (3). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.

5.13 HitMap Delay Register Test

1. Send the HitMap_Delay_Wr command with a data argument of 5'h0 (0). Send the HitMap_Delay_Rd command and read back this commanded data argument.
2. Send the HitMap_Delay_Wr command with a data argument of 5'h15 (21). Send the HitMap_Delay_Rd command and read back this commanded data argument.
3. Send the HitMap_Delay_Wr command with a data argument of 5'h0A (10). Send the HitMap_Delay_Rd command and read back this commanded data argument.

4. Send the HitMap_Delay_Wr command with a data argument of 5'h1F (31). Send the HitMap_Delay_Rd command and read back this commanded data argument.
5. Send the HitMap_Delay_Wr command with a data argument of 5'h10 (16). Send the HitMap_Delay_Rd command and read back this commanded data argument.

5.14 PHA En0 Register Test

1. Send the PHA_En0_Wr command with a data argument of 16'h0 (0). Send the PHA_En0_Rd command and read back this commanded data argument.
2. Send the PHA_En0_Wr command with a data argument of 16'h5555 (21845). Send the PHA_En0_Rd command and read back this commanded data argument.
3. Send the PHA_En0_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_En0_Rd command and read back this commanded data argument.
4. Send the PHA_En0_Wr command with a data argument of 16'hFFFF (65535). Send the PHA_En0_Rd command and read back this commanded data argument.

5.15 Veto En0 Register Test

1. Send the VETO_En0_Wr command with a data argument of 16'h0 (0). Send the VETO_En0_Rd command and read back this commanded data argument.
2. Send the VETO_En0_Wr command with a data argument of 16'h5555 (21845). Send the VETO_En0_Rd command and read back this commanded data argument.
3. Send the VETO_En0_Wr command with a data argument of 16'hAAAA (43690). Send the VETO_En0_Rd command and read back this commanded data argument.
4. Send the VETO_En0_Wr command with a data argument of 16'hFFFF (65535). Send the VETO_En0_Rd command and read back this commanded data argument.

5.16 PHA En1 Register Test

1. Send the PHA_En1_Wr command with a data argument of 2'h0 (0). Send the PHA_En1_Rd command and read back this commanded data argument.
2. Send the PHA_En1_Wr command with a data argument of 2'h1 (1). Send the PHA_En1_Rd command and read back this commanded data argument.
3. Send the PHA_En1_Wr command with a data argument of 2'h2 (2). Send the PHA_En1_Rd command and read back this commanded data argument.
4. Send the PHA_En1_Wr command with a data argument of 2'h3 (3). Send the PHA_En1_Rd command and read back this commanded data argument.

5.17 Veto En1 Register Test

1. Send the VETO_En1_Wr command with a data argument of 2'h0 (0). Send the VETO_En1_Rd command and read back this commanded data argument.
2. Send the VETO_En1_Wr command with a data argument of 2'h1 (1). Send the VETO_En1_Rd command and read back this commanded data argument.

3. Send the VETO_En1_Wr command with a data argument of 2'h2 (2). Send the VETO_En1_Rd command and read back this commanded data argument.
4. Send the VETO_En1_Wr command with a data argument of 2'h3 (3). Send the VETO_En1_Rd command and read back this commanded data argument.

5.18 MaxPHA Register Test

1. Send the MaxPHA_Wr command with a data argument of 5'h0 (0). Send the MaxPHA_Rd command and read back this commanded data argument.
2. Send the MaxPHA_Wr command with a data argument of 5'h15 (21). Send the MaxPHA_Rd command and read back this commanded data argument.
3. Send the MaxPHA_Wr command with a data argument of 5'h0A (10). Send the MaxPHA_Rd command and read back this commanded data argument.
4. Send the MaxPHA_Wr command with a data argument of 5'h1F (31). Send the MaxPHA_Rd command and read back this commanded data argument.
5. Send the MaxPHA_Wr command with a data argument of 5'h4 (4). Send the MaxPHA_Rd command and read back this commanded data argument.

5.19 GARC Mode Register Test

1. Send the GARC_Mode_Wr command with a data argument of 11'h2 (2). Send the GARC_Mode_Rd command and read back this commanded data argument.
2. Send the GARC_Mode_Wr command with a data argument of 11'h4 (4). Send the GARC_Mode_Rd command and read back this commanded data argument.
3. Send the GARC_Mode_Wr command with a data argument of 11'h8 (8). Send the GARC_Mode_Rd command and read back this commanded data argument.
4. Send the GARC_Mode_Wr command with a data argument of 11'h10 (16). Send the GARC_Mode_Rd command and read back this commanded data argument.
5. Send the GARC_Mode_Wr command with a data argument of 11'h20 (32). Send the GARC_Mode_Rd command and read back this commanded data argument.
6. Send the GARC_Mode_Wr command with a data argument of 11'h40 (64). Send the GARC_Mode_Rd command and read back this commanded data argument.
7. Send the GARC_Mode_Wr command with a data argument of 11'h80 (128). Send the GARC_Mode_Rd command and read back this commanded data argument.
8. Send the GARC_Mode_Wr command with a data argument of 11'h100 (256). Send the GARC_Mode_Rd command and read back this commanded data argument.
9. Send the GARC_Mode_Wr command with a data argument of 11'h200 (512). Send the GARC_Mode_Rd command and read back this commanded data argument.
10. Send the GARC_Mode_Wr command with a data argument of 11'h400 (1024). Send the GARC_Mode_Rd command and read back this commanded data argument.
11. Send the GARC_Mode_Wr command with a data argument of 11'h300 (768). Send the GARC_Mode_Rd command and read back this commanded data argument.

5.20 PHA Threshold Channel 00 Register Test

1. Send the PHA_Thresh00_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh00_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh00_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh00_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh00_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh00_Rd command and read back this commanded data argument.

5.21 PHA Threshold Channel 01 Register Test

1. Send the PHA_Thresh01_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh01_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh01_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh01_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh01_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh01_Rd command and read back this commanded data argument.

5.22 PHA Threshold Channel 02 Register Test

1. Send the PHA_Thresh02_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh02_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh02_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh02_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh02_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh02_Rd command and read back this commanded data argument.

5.23 PHA Threshold Channel 03 Register Test

1. Send the PHA_Thresh03_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh03_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh03_Rd command and read back this commanded data argument.

3. Send the PHA_Thresh03_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh03_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh03_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh03_Rd command and read back this commanded data argument.

5.24 PHA Threshold Channel 04 Register Test

1. Send the PHA_Thresh04_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh04_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh04_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh04_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh04_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh04_Rd command and read back this commanded data argument.

5.25 PHA Threshold Channel 05 Register Test

1. Send the PHA_Thresh05_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh05_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh05_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh05_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh05_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh05_Rd command and read back this commanded data argument.

5.26 PHA Threshold Channel 06 Register Test

1. Send the PHA_Thresh06_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh06_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh06_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh06_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh06_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh06_Rd command and read back this commanded data argument.

5.27 PHA Threshold Channel 07 Register Test

1. Send the PHA_Thresh07_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh07_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh07_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh07_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh07_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh07_Rd command and read back this commanded data argument.

5.28 PHA Threshold Channel 08 Register Test

1. Send the PHA_Thresh00_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh00_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh00_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh00_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh00_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh00_Rd command and read back this commanded data argument.

5.29 PHA Threshold Channel 09 Register Test

1. Send the PHA_Thresh09_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh09_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh09_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh09_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh09_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh09_Rd command and read back this commanded data argument.

5.30 PHA Threshold Channel 10 Register Test

1. Send the PHA_Thresh10_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh10_Rd command and read back this commanded data argument.

2. Send the PHA_Thresh10_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh10_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh10_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh10_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh10_Rd command and read back this commanded data argument.

5.31 PHA Threshold Channel 11 Register Test

1. Send the PHA_Thresh11_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh11_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh11_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh11_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh11_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh11_Rd command and read back this commanded data argument.

5.32 PHA Threshold Channel 12 Register Test

1. Send the PHA_Thresh12_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh12_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh12_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh12_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh12_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh12_Rd command and read back this commanded data argument.

5.33 PHA Threshold Channel 13 Register Test

1. Send the PHA_Thresh13_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh13_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh13_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh13_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh13_Rd command and read back this commanded data argument.

5. Send the PHA_Thresh13_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh13_Rd command and read back this commanded data argument.

5.34 PHA Threshold Channel 14 Register Test

1. Send the PHA_Thresh14_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh14_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh14_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh14_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh14_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh14_Rd command and read back this commanded data argument.

5.35 PHA Threshold Channel 15 Register Test

1. Send the PHA_Thresh15_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh15_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh15_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh15_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh15_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh15_Rd command and read back this commanded data argument.

5.36 PHA Threshold Channel 16 Register Test

1. Send the PHA_Thresh16_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh16_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh16_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh16_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh16_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh16_Rd command and read back this commanded data argument.

5.37 PHA Threshold Channel 17 Register Test

1. Send the PHA_Thresh17_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh17_Wr command with a data argument of 12'h555 (1365). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh17_Wr command with a data argument of 12'hAAA (2730). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh17_Wr command with a data argument of 12'hFFF (4095). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh17_Wr command with a data argument of 12'h45A (1114). Send the PHA_Thresh17_Rd command and read back this commanded data argument.

5.38 ADC TACQ Register Test

1. Send the ADC_TACQ_Wr command with a data argument of 6'h0 (0). Send the ADC_TACQ_Rd command and read back this commanded data argument.
2. Send the ADC_TACQ_Wr command with a data argument of 6'h15 (21). Send the ADC_TACQ_Rd command and read back this commanded data argument.
3. Send the ADC_TACQ_Wr command with a data argument of 6'h2A (42). Send the ADC_TACQ_Rd command and read back this commanded data argument.
4. Send the ADC_TACQ_Wr command with a data argument of 6'h3F (63). Send the ADC_TACQ_Rd command and read back this commanded data argument.
5. Send the ADC_TACQ_Wr command with a data argument of 6'h0 (0). Send the ADC_TACQ_Rd command and read back this commanded data argument.
6. Send the GARC_Reset command to ensure all registers are at the proper initial value.

At the successful conclusion of this section, the GARC commandable register bits have been demonstrated to be functional with all bits toggling as commanded.

5.39 GAFE ASICs Mode Register Test

This section tests the proper functioning of the GAFE mode register. Repeat for all GAFE ASICs present using valid GAFE addressing.

1. Send the GAFE_Config_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_Config_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_Config_Write command to the GAFE being tested with a data field of 16'hFF (255).
4. Send the GAFE_Config_Read command. The data field in the return data stream should be 16'hFF (255).
5. Send the GAFE_Config_Write command to the GAFE being tested with a data field of 16'h30 (48).
6. Send the GAFE_Config_Read command. The data field in the return data stream should be 16'h30 (48).

5.40 GAFE ASIC VETO DAC Register Test

This section tests the proper functioning of the GAFE VETO DAC register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_VETO_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_VETO_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_VETO_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_VETO_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_VETO_Write command to the GAFE being tested with a data field of 16'h39 (57).
6. Send the GAFE_VETO_Read command. The data field in the return data stream should be 16'h39 (57).

5.41 GAFE ASIC VETO VERNIER Register Test

This section tests the proper functioning of the GAFE VETO VERNIER register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_VERNIER_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_VERNIER_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_VERNIER_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_VERNIER_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_VERNIER_Write command to the GAFE being tested with a data field of 16'h26 (38).
6. Send the GAFE_VERNIER_Read command. The data field in the return data stream should be 16'h26 (38).

5.42 GAFE ASIC HLD DAC Register Test

This section tests the proper functioning of the GAFE HLD_DAC register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_HLD_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_HLD_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_HLD_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_HLD_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_HLD_Write command to the GAFE being tested with a data field of 16'h37 (55).
6. Send the GAFE_HLD_Read command. The data field in the return data stream should be 16'h37 (55).

5.43 GAFE ASIC BIAS_DAC Register Test

This section tests the proper functioning of the GAFE BIAS_DAC register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_BIAS_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_BIAS_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_BIAS_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_BIAS_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_BIAS_Write command to the GAFE being tested with a data field of 16'h20 (32).
6. Send the GAFE_BIAS_Read command. The data field in the return data stream should be 16'h20 (32).

5.44 GAFE ASIC TCI_DAC Register Test

This section tests the proper functioning of the GAFE TCI_DAC register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_TCI_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_TCI_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_TCI_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_TCI_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_TCI_Write command to the GAFE being tested with a data field of 16'h00 (0).
6. Send the GAFE_TCI_Read command. The data field in the return data stream should be 16'h00 (0).

At the successful conclusion of this register test section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

5.45 Initial GAFE Logic Reset Test

After initial power up or a GARC Reset command is sent, the GAFE registers should be initialized. This is tested by the following command sequence (sent to each GAFE being tested at the unique five bit address for that GAFE). This test may be automated using the LabView GSE via the **GAFE LOGIC RESET TEST.TXT** script.

1. Send the GARC_Reset command.
2. Send the GAFE_Config_Read command. The data field in the return data stream should be 16'h30 (48).
3. Send the GAFE_Version Command. The GAFE Version indicates the version of the GAFE logic core and should be 3.
4. Send the GAFE_VETO_Read command. The data field in the return data stream should be 16'h39 (57).
5. Send the GAFE_VERNIER_Read command. The data field in the return data stream should be 16'h26 (38).

6. Send the GAFE_HLD_Read command. The data field in the return data stream should be 16'h37 (55).
7. Send the GAFE_BIAS_Read command. The data field in the return data stream should be 16'h20 (32).
8. Send the GAFE_TCI_Read command. The data field in the return data stream should be 16'h00 (0).
9. Send the GAFE_Version command. The data field in the return data stream should be 16'h03 (3).
10. Send the GAFE_Write_Ctr command. The data field in the return data stream should be 16'h00 (0).
11. Send the GAFE_Reject_Ctr command. The data field in the return data stream should be 16'h00 (0).
12. Send the GAFE_Cmd_Ctr command. The data field in the return data stream should be 16'h0A (10).
13. Send the GAFE_Chip_Address command for each of the 18 GAFE's. The data field in the return data stream should be the address of the chip that was commanded.
14. At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

5.46 GAFE ASIC Broadcast Command Functional Verification

This section tests the proper response to broadcast commands (e.g., address 31) by the GAFEs.

This test may be automated using the LabView GSE via the **GAFE Broadcast Command Test.txt** script.

1. Send the GARC_Reset command.
2. Send the GAFE_Config_Read command with a broadcast address, i.e., address 'h1F (31) , to all the GAFEs being tested. There should be no data returned from any GAFE.
3. Send the GAFE_Config_Write command with a broadcast address and a data field of 'hAA (170) to the GAFEs.
4. Send a GAFE_Config_Read address command to each GAFE under test. The return data field from each addressed GAFE should be 'hAA (170).
5. Send a GARC_Reset command to reinitialize each of the GAFE logic cores.
6. At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

5.47 Test of the GARC Parity Error Detection and Error Generation

This section tests the proper functioning of the GARC return data parity select bit and the GARC's ability to detect command and data parity errors.

This test may be automated using the LabView GSE via the **GARC Parity Test.txt** script.

1. Send the GARC_Reset command.
2. Send the GARC_Status command. The data field in the return data stream should be decimal 24.
3. Send the GARC_Mode command with a data argument of decimal 769, a command to return event data with even parity.

4. Send the GARC_Status command. The data field in the return data stream should show an AEM parity error.
5. Send the GARC_Mode command with a data argument of decimal 768, a command back to odd parity, the nominal mode.
6. Send the GARC_Status command. The data field in the return data stream should be decimal 24. The AEM should indicate nominal parity (no errors).
7. Send the GARC_Reset command to reinitialize all GARC registers. Ready to test Command and Data parity errors.
8. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b0000.
9. Send the GARC_Version command.
10. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b0000.
11. Send the Command_Register command and verify that the return data is 16'h0000.
12. Send the even command parity (i.e., error) command: 34'h2404C0001.
13. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b1101.
14. Send the Command_Register command and verify that the return data is 16'h404E.

At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

5.48 Test of the Look-At-Me Circuitry

This section tests the proper functioning of the GARC Look-At-Me circuitry. The GARC has the capability to receive commands from either the primary side or secondary side. The selection of which set of receivers to listen to is controlled by the Look-At-Me circuitry. This circuitry toggles the status of the receiving side based upon receipt of a special command pattern (e.g., **34'h24153D721**). This the equivalent of sending a GARC configuration command to addr 1, function 4, with a data pattern of 60304.

Another way to look at the bit pattern (in a format like the ICD would present it) would be:

1001	==	configuration command
0	==	GARC
00001	==	address 1
0	==	write
1	==	a command with data
0100	==	function 4
1	==	command parity
16'h	==	EB90 == data field (60304)
0	==	data parity

This test may be automated using the LabView GSE via the **GARC Look At Me Test.txt** script.

Turn on the system and send a Look-at-me command. Send the GARC_Status command. The data field in the return data stream should be decimal 24. Bit 0 (LSB) of the Status register represents the Look-At-Me status (0 = A, 1 = B).. Set the GSE to the Secondary side, Send the GARC_Status command. Observe that there is no response from the GARC.

Send the Look_At_Me command to the GARC from the secondary side interface. Send the GARC_Status command. The data field in the return data should be decimal 25 (with the LSB = 1, indicating the Look-At-Me status is B side). Set the GSE for the Primary side and send the GARC_Status command. Observe that

there is no response from the GARC. Send the Look_At_Me command to the GARC from the primary side interface. Send the GARC_Status command. The data field in the return data should be decimal 24 (with the LSB = 0, indicating the Look-At-Me status is A side).

This concludes the test of the GARC Look-At-Me circuitry. At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

5.49 Maximum PHA Return Test

This section tests the proper functioning of the event data processor as it handles the Maximum Number of PHA words command.

This test may be automated using the LabView GSE via the **GARC_Max_PHA_Return_Test.tst** script.

1. Send the Max_PHA_Wr command with a data argument of 18. Send the Max_PHA_Rd command and verify that the data value has been set. Send the Trigger_ZS command and verify that the event data processor sends back 18 PHA words.
2. Repeat the previous step with a data argument of 17.
3. Repeat the previous step with a data argument of 16.
4. Repeat the previous step with a data argument of 15.
5. Repeat the previous step with a data argument of 14.
6. Repeat the previous step with a data argument of 13.
7. Repeat the previous step with a data argument of 12.
8. Repeat the previous step with a data argument of 11.
9. Repeat the previous step with a data argument of 10.
10. Repeat the previous step with a data argument of 9.
11. Repeat the previous step with a data argument of 8.
12. Repeat the previous step with a data argument of 7.
13. Repeat the previous step with a data argument of 6.
14. Repeat the previous step with a data argument of 5.
15. Repeat the previous step with a data argument of 4.
16. Repeat the previous step with a data argument of 3.
17. Repeat the previous step with a data argument of 2.
18. Repeat the previous step with a data argument of 1.
19. Repeat the previous step with a data argument of 0.
20. Send the GARC_Reset command to restore the Max_PHA number to the default. This completes the test of the GARC MAX_PHA function. At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

5.50 PHA Enable/Disable Test

This section tests the proper functioning of the event data processor as it handles the PHA enables and disables in selection of PHA words for transmission.

This test may be automated using the LabView GSE via the GARC PHA Enable Test.tst script.

1. Send the PHA_En0_Wr command with a data argument of decimal 'hFFFF (65535). Verify this value with the PHA_En0_Rd command. This enables channels 0 – 15.
2. Send the PHA_En1_Wr command with a data argument of decimal 3. Verify this value with the PHA_En1_Rd command. This enables channels 16 and 17.
3. Send the Trigger_NOZS command. The event data processor should respond with 18 PHA words during event readout.
4. Send the PHA_En0_Wr command with a data argument of 'hFFFE (65534). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 0.
5. Send the PHA_En0_Wr command with a data argument of 'hFFFD (65533). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 1.
6. Send the PHA_En0_Wr command with a data argument of 'hFFFB (65531). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 2.
7. Send the PHA_En0_Wr command with a data argument of 'hFFF7 (65527). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 3.
8. Send the PHA_En0_Wr command with a data argument of 'hFFEF (65519). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 4.
9. Send the PHA_En0_Wr command with a data argument of 'hFFDF (65503). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 5.
10. Send the PHA_En0_Wr command with a data argument of 'hFFBF (65471). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 6.
11. Send the PHA_En0_Wr command with a data argument of 'hFF7F (65407). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 7.
12. Send the PHA_En0_Wr command with a data argument of 'hFEFF (65279). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 8.
13. Send the PHA_En0_Wr command with a data argument of 'hFDFF (65023). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 9.
14. Send the PHA_En0_Wr command with a data argument of 'hFBFF (64511). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 10.

15. Send the PHA_En0_Wr command with a data argument of 'hF7FF (63487). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 11.
16. Send the PHA_En0_Wr command with a data argument of 'hEFFF (61439). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 12.
17. Send the PHA_En0_Wr command with a data argument of 'hDFFF (57343). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 13.
18. Send the PHA_En0_Wr command with a data argument of 'hBFFF (49151). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 14.
19. Send the PHA_En0_Wr command with a data argument of 'h7FFF (32767). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 15.
20. Send the PHA_En0_Wr command with a data argument of 'hFFFF (65535). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words.
21. Send the PHA_En1_Wr command with a data argument of 2. Verify this value with the PHA_En1_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 16.
22. Send the PHA_En1_Wr command with a data argument of 1. Verify this value with the PHA_En1_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 17.
23. Send the GARC_Reset command to reinitialize all GARC registers. This completes the testing of the PHA enable and disable register bits. At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

5.51 PHA Threshold Verification Test

This section tests the proper functioning of each of the 18 PHA thresholding circuits. The intent of the circuits is such that the ZS Map bit is set active high for a given PHA word if either the range bit is high or the 12 bit PHA word is greater than the PHA threshold.

This test may be automated using the LabView GSE via the **GARC PHA Threshold Test.txt** script.

1. Send the GARC_Reset command.
2. Send the Max_PHA_Wr command with a data argument of 18.
3. Send the GAFE Config Wr command with a data argument of 50 (enable TCI) to channel 0.
4. Send the GAFE TCI Wr command with a data argument of 10 to channel 0.
5. Send the PHA Thresh00 Wr command with a data argument of 100.
6. Repeat these steps for channels 01 through 17.
7. Send the TRIG_ZS command. All 18 channels of PHA should be returned.
8. Send the PHA_Thresh00_Wr command with a data argument of 3000. Verify with the PHA_Thresh00_Rd command.
9. Send the TRIG_ZS command. All PHA channels except for channel 0 should be returned in the event data.

10. Send the PHA_Thresh01_Wr command with a data argument of 3000 and set PHA Thresh00 back to 100. Verify with the PHA_Thresh_Rd command.
11. Send the TRIG_ZS command. All PHA channels except for channel 1 should be returned in the event data.
12. Repeat the above two steps serially for channels 2 – 17, verifying in each case that the proper event word is zero suppressed in the event data.
13. Send the GARC_Reset command to restore all values to the power-on defaults.

This completes the PHA Threshold Verification section of the procedure. Record the successful completion of this test in the Test Record at the end of this procedure.

5.52 Test of the HVBS DAC Commands, DAC Buffer, and Differential Drivers

To verify the operation of the HVBS DAC and Differential Driver make the following measurements. Verify that HVBS DAC is set to 0.

The reference for these measurements is the Lepracon connector shell.

The HVBS_Level_Wr command and Use_HV_Nominal_Wr command to change the DAC setting as shown to verify the linearity of the circuit.

This test may be automated using the LabView GSE via the **GARC HVBS DAC Level Test.txt** script.

Meas. No.	DAC SETTING	JHV1-4 (HV-DACP)	JHV1-9 (HV-DACN)
1	000		
2	200		
3	400		
4	600		
5	800		
6	1000		
7	1200		
8	1400		
9	1600		
10	1800		
11	2000		
12	2200		
13	2400		
14	2600		
15	2800		
16	3000		
17	3200		
18	3400		
19	3600		
20	3800		
21	4000		
22	4095		

Meas. No.	DAC SETTING	JHV2-4 (HV-DACP)	JHV2-9 (HV-DACN)
1	000		
2	4095		

5.53 Test of the HVBS Triple Modular Redundancy Circuitry

This section tests the proper functioning of the GARC HVBS enable circuitry. Each pattern in the TMR logic will be tested to verify proper recovery from a single event upset. The GARC HV_ENABLE_1 pin is 185 and the HV_ENABLE_2 pins is 186. A truth table for proper TMR circuitry function is detailed below.

This test may be automated using the LabView GSE via the GARC HV Enable Test.txt script.

Enable Bit A	Enable Bit B	Enable Bit C	HV Enable Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

To verify the Enable command connect volt meters at JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) of the FREE board. Exact voltages for command on and off will be measured at the end of this section. For nominal power supply voltages, the nominal “ON” voltage is approximately 3.2V and “OFF” is approximately 0.015V. Reference the meter to the Lepracon connector body.

1. Send the GARC_Mode_Wr command with a data argument of 768 and verify the data with the GARC_Mode_Rd command.
2. Send the GARC_Status command. The data field in the return data stream should show that bits 1 and 2 are 0, indicating that HVBS 1 and 2, respectively, are disabled. The data field value should be 24. Verify that JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) are at 0V.
3. Send the GARC_Mode_Wr command with decimal argument 770 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. . The data field value should be 24. Verify that JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) are at 0V.
4. Send the GARC_Mode_Wr command with decimal argument 772 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. The data field value should be 24. Verify that JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) are at 0V.
5. Send the GARC_Mode_Wr command with decimal argument 774 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is enabled and HVBS 2 is disabled. . The data field value should be 26. Verify that JHV1-5 (HV_Enable_1) is at 3.3V and JHV2-5 (HV_Enable_2) is at 0V.

6. Send the GARC_Mode_Wr command with decimal argument 776 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. . The data field value should be 24. Verify that JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) is at 0V.
7. Send the GARC_Mode_Wr command with decimal argument 778 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is enabled and HVBS 2 is disabled. . The data field value should be 26. Verify that JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) is at 0V.
8. Send the GARC_Mode_Wr command with decimal argument 780 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is enabled and HVBS 2 is disabled. The data field value should be 26. Verify that JHV1-5 (HV_Enable_1) is at 3.3V and JHV2-5 (HV_Enable_2) is at 0V.
9. Send the GARC_Mode_Wr command with decimal argument 782 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is enabled and HVBS 2 is disabled. The data field value should be 26. Verify that JHV1-5 (HV_Enable_1) is at 3.3V and JHV2-5 (HV_Enable_2) is at 0V.
10. Send the GARC_Mode_Wr command with decimal argument 784 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. The data field value should be 24. Verify that JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) is at 0V.
11. Send the GARC_Mode_Wr command with decimal argument 800 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. The data field value should be 24. Verify that JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) is at 0V.
12. Send the GARC_Mode_Wr command with decimal argument 816 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is enabled. The data field value should be 28. Verify that JHV1-5 (HV_Enable_1) is at 0V and JHV2-5 (HV_Enable_2) is at 3.3V.
13. Send the GARC_Mode_Wr command with decimal argument 832 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. The data field value should be 24. Verify that JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) is at 0V.
14. Send the GARC_Mode_Wr command with decimal argument 848 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is enabled. The data field value should be 28. Verify that JHV1-5 (HV_Enable_1) is at 0V and JHV2-5 (HV_Enable_2) is at 3.3V.

15. Send the GARC_Mode_Wr command with decimal argument 864 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is enabled. The data field value should be 28. Verify that JHV1-5 (HV_Enable_1) is at 0V and JHV2-5 (HV_Enable_2) is at 3.3V.
16. Send the GARC_Mode_Wr command with decimal argument 880 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is enabled. The data field value should be 28. Verify that JHV1-5 (HV_Enable_1) is at 0V and JHV2-5 (HV_Enable_2) is at 3.3V.
17. Send the GARC_Mode_Wr command with decimal argument 782 and verify the data with the GARC_Mode_Rd command. Measure the voltage at the two HV_ENABLE pins:
 - a. HV_ENABLE_1 voltage JHV1-5: _____ (expected ~ 3.2V)
 - b. HV_ENABLE_2 voltage JHV2-5: _____ (expected ~ 3.2V)
18. Send the GARC_Mode_Wr command with decimal argument 768 and verify the data with the GARC_Mode_Rd command. Measure the voltage at the two HV_ENABLE pins:
 - a. HV_ENABLE_1 voltage JHV1-5: _____ (expected ~ 0.015V)
 - b. HV_ENABLE_2 voltage JHV2-5: _____ (expected ~ 0.015V)
19. Send the GARC_Reset command. Verify that JHV1-5 (HV_Enable_1) and JHV2-5 (HV_Enable_2) is at 0V.

5.54 Test of the SAA/HV Normal Modes

This section tests the proper functioning of the SAA/HV Normal modes.

1. Send the HVBS_Level_Wr command with a data argument of decimal 2048. Send the HVBS_Level_Rd command to verify the write command.
2. Send the SAA_Level_Wr command with a data argument of decimal 1024. Send the SAA_Level_Rd command to verify.
3. Using a multimeter at the JHV1 pin 4 connector, verify the HV_DACP voltage is 1.4V.
Value observed: _____
4. Send the Use_HV_Nominal command. Verify that the HV_DACP voltage is 2.1V.
Value observed: _____
4. Send the DAC_HVReg_Rd command and verify the return data pattern from the MAX5121 DAC is decimal 10240.
Value observed: _____
5. Send the Use_SAA_Level command. Verify that the HV_DACP voltage is 1.8V.
Value observed: _____
6. Send the DAC_SAAREg_Rd command and verify the return data pattern from the MAX5121 DAC is decimal 9216.
Value observed: _____

7. Send the GARC_Reset command. Verify that the HV_DACP voltage is 1.4V.

Value observed: _____

9. At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

5.55 Test of the GARC LVDS Circuitry Driver Currents

This test may be omitted at the discretion of the Test Conductor.

This section tests the proper functioning of the GARC LVDS Driver circuitry. It is required that all LVDS drivers be terminated at the receiver with a 100 ohm resistor. The ACD-to-AEM nominal drive current is 3.5 mA across the 100 ohms for a voltage differential of 350 mV.

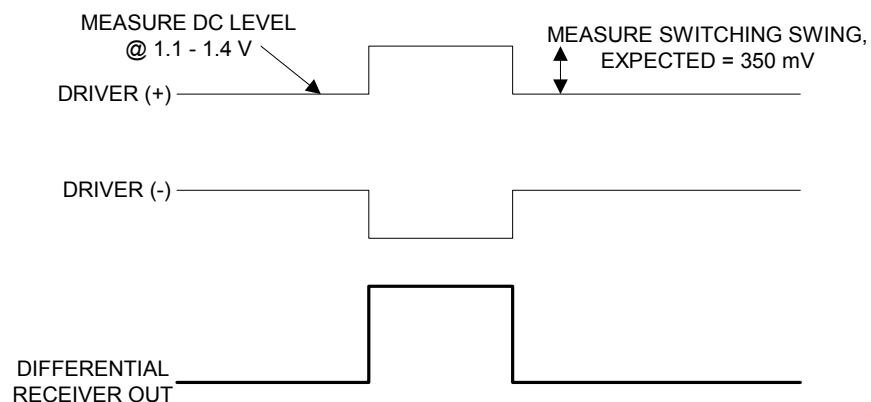
There are 6 LVDS receiver input pairs to the GARC:

- a) ACD_CLK_A, ACD_CLK_B
- b) ACD_CMD_A, ACD_CMD_B
- c) ACD_RESET_A, ACD_RESET_B

There are 40 LVDS driver output pairs from the GARC:

- d) ACD_NSDATA_A, ACD_NSDATA_B
- e) ACD_NVETO_nnA, ACD_NVETO_nnB (where nn is 00 – 17)
- f) ACD_CNO_A, ACD_CNO_B

The “A” side interface will be examined first. Verify that an AEM (or AEM simulator) is connected to the GARC “A” side interface. Using a digitizing oscilloscope, measure the DC baseline and switching differential voltage on each of the driver/receiver pairs. Two scope probes may be used, one on each side of the differential receiver inputs. The DC level and switching voltages may be measured as diagrammed below. The measurement will be made with the Drivers enabled and with them disabled.



Remove the Breakout Box on JP1 and JS2 and record the information for each pin on the AEM Simulator in the table below. Please document the data from the following steps in the copy of the Comprehensive Performance Test Record associated with this procedure.

Obtain a scope picture of the nominal pulse Amplitude Swings

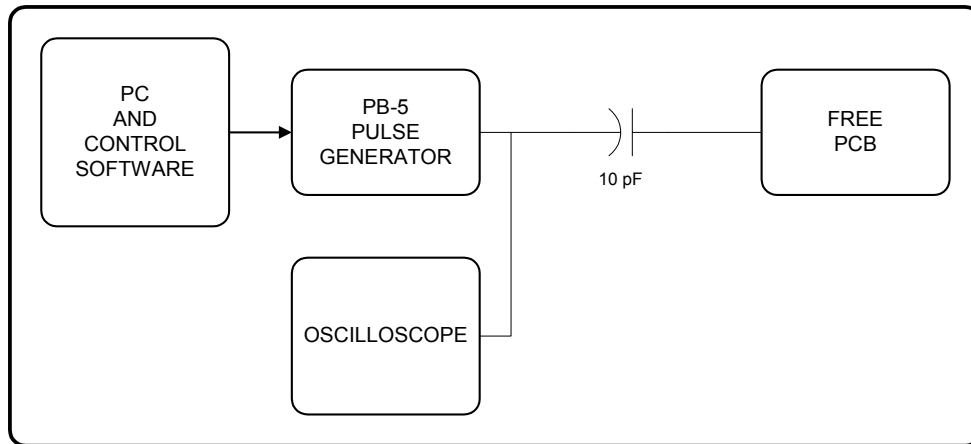
GARC Signal	Signal Location	- DC Level (V) Enabled	+ DC Level (V) Enabled	- DC Level (V) Disabled	+ DC Level (V) Disabled
ACD NVETO 12B	R1				
ACD NVETO 12A	R2				
ACD NVETO 11B	R3				
ACD NVETO 11A	R4				
ACD NVETO 10B	R5				
ACD NVETO 10A	R6				
ACD NVETO 09B	R7				
ACD NVETO 09A	R8				
ACD NVETO 08B	R9				
ACD NVETO 08A	R10				
ACD NVETO 07B	R11				
ACD NVETO 07A	R12				
ACD NVETO 06B	R13				
ACD NVETO 06A	R14				
ACD NVETO 05B	R15				
ACD NVETO 05A	R16				
ACD NVETO 04B	R17				
ACD NVETO 04A	R18				
ACD NVETO 03B	R19				
ACD NVETO 03A	R20				
ACD NVETO 02B	R21				
ACD NVETO 02A	R22				
ACD NVETO 01B	R23				
ACD NVETO 01A	R24				
ACD NVETO 00B	R25				
ACD NVETO 00A	R26				
ACD CNO B	R27				
ACD CNO A	R28				
ACD NVETO 13B	R31				
ACD NVETO 13A	R32				
ACD NVETO 14B	R33				
ACD NVETO 14A	R34				
ACD NVETO 15B	R35				
ACD NVETO 15A	R36				
ACD NVETO 16B	R37				
ACD NVETO 16A	R38				
ACD NVETO 17B	R39				
ACD NVETO 17A	R40				
ACD NSDATA B	R29				
ACD NSDATA A	R30				

5.56 Characterization of each of the 18 GAFE chips

This test may be omitted at the discretion of the Test Conductor.

The following test utilizes a NIM-mounted laboratory pulse generator to simulate a phototube input at the GAFE inputs. These tests will characterize the GAFE ASICs and associated readout circuitry. This test will also verify the operation for each GAFE ASIC on the FREE printed circuit card. Included in this test will be the ADC conversion values as measured at the GARC. The low-energy threshold will be characterized during this test. The HLD enable/disable function will be tested.

A 10 pF charge terminator will be used for the each FREE board channel in the following setup.



In this case, the charge input to the FREE card is $Q = C \cdot V$, so 0.64 pC is approximately 64 mV on the scope.

Run the automatic characterization script to test the following

- 1) Register test
- 2) Bias DAC Test (LE)
- 3) Bias DAC Test (HE)
- 4) Hold Delay
- 5) INL Test (LE)
- 6) INL Test (HE)
- 7) Crossover point (LE/HE)
- 8) Hitmap Delay Test
- 9) Veto DAC Test
- 10) Vernier DAC Test
- 11) HLD DAC Test
- 12) TCI (LE)
- 13) TCI (HE)

All printouts are to be attached to the Test Results Record.

Settings being used for the following tests:

Create a configuration script for the FREE board under test.

File Name _____

GAFE Configuration

GAFE Channel	Mode	VETO DAC	VETO VERNIER	HLD DAC	BIAS DAC	TCI DAC
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						

GARC Configuration

Register	Test Value
Veto_Delay	
HVBS_Level	
SAA_Level	
Hold_Delay	
Veto_Width	
HitMap_Width	
HitMap_Deptime	
HitMap_Delay	
PHA_En0	
PHA_En1	
VETO_En0	
VETO_En1	
Max_PHA	

GARC_Mode	
FREE_Board_ID	
GARC_Version	
PHA_Threshold_00	
PHA_Threshold_01	
PHA_Threshold_02	
PHA_Threshold_03	
PHA_Threshold_04	
PHA_Threshold_05	
PHA_Threshold_06	
PHA_Threshold_07	
PHA_Threshold_08	
PHA_Threshold_09	
PHA_Threshold_10	
PHA_Threshold_11	
PHA_Threshold_12	
PHA_Threshold_13	
PHA_Threshold_14	
PHA_Threshold_15	
PHA_Threshold_16	
PHA_Threshold_17	
ADC_TACQ	

5.57 TCI and GAFE ADC Checkout

This test may be Omitted at the discretion of the Test Conductor.

This test will record the ADC output for each GAFE channel with several TCI settings. Run the TCI VI from the Labview GSE and attach the printouts to the test record sheet. Record the File name for the data.

File Name - _____

5.58 Power Supply Rail Tests – 3.0V to 3.6V

Power supply rail voltage testing has been performed on the FREE card assembly. This test will verify the operation of the card at the upper and lower voltage limits of +3.60V and +3.00V, respectively.

Set the power supply to +3.6V.

Run the register test from section 5.5

Run the test of the DAC interface circuitry, section 5.52

Run the characterization test from 5.5 or the TCI Test from section 5.57

Record the Data File name - _____

Set the power supply to +3.0V.

Run the register test from section 5.5

Run the test of the DAC interface circuitry, section 5.52

Run the characterization test from 5.5 or the TCI Test from section 5.57

Record the Data File name - _____

Appendix 1: GARC Documentation

A more complete set of documentation (such as the Verilog, EDIF, layout, and wirebonding diagram) is available on the LHEA ACD electronics web page at:

<http://lhea-glast.gsfc.nasa.gov/acd/electronics/#garc>

Appendix 2: GARC Configuration Command Format

The GARC logic core responds only to properly structured commands. There are two possible command types – Trigger Commands and Configuration Commands. Trigger commands initiate an Event Data cycle, causing a GAFE Hold, an analog-to-digital conversion, and the return of an event data packet. Configuration commands are used to either read or write GARC or GAFE registers.

The format for Trigger Commands is detailed in the table below.

Bit(s)	Bit Description	Value
3	Start Bit	1
2:1	Trigger Type Bits	10 for ZS Enabled Trigger 01 for Send All PHA Trigger
0	Parity Bit	Odd parity bit over previous two bits

Therefore, a 1100 is a ZS Enabled Trigger command and a 1010 is a Send All PHA Trigger command.

This format for GARC Configuration Commands is detailed in the table below.

Field	# bits	Function
Start	1	1 for start
CMD Type	2	00 for command
CMD Type Parity	1	Odd Parity over previous 2 bits (without Start bit)
GAFE/GARC Select	1	0 for GARC 1 for GAFE
GAFE/GARC Address	5	GAFE: Select which GAFE, 0x1F for all GAFE GARC: Select which function block
Read/Write	1	0 for write, 1 for read
Data/Dataless	1	0 for dataless, 1 for data, always 1 for ACD
register/function number	4	Which register/function in the function block
CMD Parity	1	Odd parity bit over previous 15 bits
Data	16	Data Field
Data Parity	1	Odd parity bit over previous 16 bits

Appendix 3: GARC Event Data Format

The GARC core returns an event data packet when a valid trigger command is received. The event data format is detailed in the table below.

<u>Field</u>	<u># bits</u>	<u>Function</u>
Start Bit	1	1 for start
Hit Map Bits	18	Bits 17-0 for channels 0-17, bit set if hit in channel
Zero Suppression Bits	18	Bits 17-0 for channels 0-17, bit set if PHA above threshold
CMD/Data ERROR	1	Error in command parity detected
Header Parity	1	Odd parity bit over previous 37 bits
PHA Words (quantity 0-18) Order: Channel 0 to channel 17	15	Bit 14: 1 if another PHA word follows this one, 0 if this is last one Bit 13: 1 for high range, 0 for low range Bits 12-1: the PHA value, 0 to 4095 Bit 0: Odd parity over last 14 bits

Appendix 4: GARC Configuration Data Readback Format

The GARC core returns an register configuration data packet when a valid configuration readback command is received. The configuration readback data format is detailed in the table below.

<u>Field</u>	<u># bits</u>	<u>Function</u>
Start	1	1 for start
GAFE/GARC Select	1	Copy of write command field (0 for GARC, 1 for GAFE)
GAFE/GARC Address	5	Copy of write command field (Select which GAFE)
Read/Write	1	1 for read
Data/Dataless	1	always 1
Register/function number	4	Copy of write command field (which register/function in the function block)
CMD Parity	1	Odd parity bit over previous 12 bits
Data	16	Data, MSB first
CMD/DATA ERROR	1	Error in parity detected
Parity	1	Odd parity bit over previous 17 bits

Appendix 5: MDM Connector Pin List**Connector JHV1 to High Voltage Bias Supply #1**

Connector Pin	Signal Name
JHV1 - 1	HVBS +28V
JHV1 - 2	28V RTN
JHV1 - 3	HV_MON_P1
JHV1 - 4	DAC_P
JHV1 - 5	HV_ENABLE_1
JHV1 - 6	HVBS +28V
JHV1 - 7	28V RTN
JHV1 - 8	HV_MON_N1
JHV1 - 9	DAC_N

Connector JHV2 to High Voltage Bias Supply #2

Connector Pin	Signal Name
JHV2 - 1	HVBS +28V
JHV2 - 2	28V RTN
JHV2 - 3	HV_MON_P2
JHV2 - 4	DAC_P
JHV2 - 5	HV_ENABLE_2
JHV2 - 6	HVBS +28V
JHV2 - 7	28V RTN
JHV2 - 8	HV_MON_N2
JHV2 - 9	DAC_N

Appendix 6: Series II Circular Connector Pin List**Pin Locations for Connectors JP1 and (JS2)**

Signal Name	Function	Connector Pin
ACD_VDD_0A(B)	+3.3V Power to ACD	1
ACD_VDD_1A(B)	+3.3V Power to ACD	3
ACD_VDD_2A(B)	+3.3V Power to ACD	4
ACD_28V_0A(B)	+28V Power to ACD HVBS	5
ACD_28V_1A(B)	+28V Power to ACD HVBS	7
ACD_NVETO_16A(B)P	Veto Ch 16 + to AEM	17
ACD_NVETO_16A(B)M	Veto Ch 16 - to AEM	18
ACD_NVETO_17A(B)P	Veto Ch 17 + to AEM	19
ACD_NVETO_17A(B)M	Veto Ch 17 - to AEM	20
ACD_NCNO_A(B)P	CNO + to AEM	21
ACD_NCNO_A(B)M	CNO - to AEM	22
ACD_HV_AP	High Voltage Supply 1 Monitor + to AEM	23

ACD_HV_AM	High Voltage Supply 1 Monitor - to AEM	24
ACD_TEMP_A(B)P	Temp Monitor + to AEM	25
ACD_TEMP_A(B)M	Temp Monitor - to AEM	26
ACD_HV_BP	High Voltage Supply 2 Monitor + to AEM	27
ACD_HV_BM	High Voltage Supply 2 Monitor - to AEM	28
ACD_GND_0A(B)	+3.3V Power Return	30
ACD_GND_1A(B)	+3.3V Power Return	31
ACD_GND_2A(B)	+3.3V Power Return	32
ACD_28V_RTN_0A(B)	+28V Return	33
ACD_28V_RTN_1A(B)	+28V Return	34
ACD_NVETO_15A(B)M	Veto Ch 15 - to AEM	40
ACD_NVETO_15A(B)P	Veto Ch 15 + to AEM	41
ACD_NVETO_14A(B)M	Veto Ch 14 - to AEM	42
ACD_NVETO_14A(B)P	Veto Ch 14 + to AEM	43
ACD_NVETO_13A(B)M	Veto Ch 13 - to AEM	44
ACD_NVETO_13A(B)P	Veto Ch 13 + to AEM	45
ACD_NVETO_12A(B)M	Veto Ch 12 - to AEM	46
ACD_NVETO_12A(B)P	Veto Ch 12 + to AEM	47
ACD_NVETO_11A(B)M	Veto Ch 11 - to AEM	48
ACD_NVETO_11A(B)P	Veto Ch 11 + to AEM	49
ACD_NVETO_10A(B)M	Veto Ch 10 - to AEM	50
ACD_NVETO_10A(B)P	Veto Ch 10 + to AEM	51
ACD_NVETO_09A(B)M	Veto Ch 9 - to AEM	52
ACD_NVETO_09A(B)P	Veto Ch 9 + to AEM	53
ACD_NVETO_08A(B)M	Veto Ch 8 - to AEM	54
ACD_NVETO_08A(B)P	Veto Ch 8 + to AEM	55
ACD_NVETO_07A(B)M	Veto Ch 7 - to AEM	56
ACD_NVETO_07A(B)P	Veto Ch 7 + to AEM	57
ACD_NVETO_06A(B)M	Veto Ch 6 - to AEM	58
ACD_NVETO_06A(B)P	Veto Ch 6 + to AEM	59
ACD_NVETO_05A(B)M	Veto Ch 5 - to AEM	60
ACD_NVETO_05A(B)P	Veto Ch 5 + to AEM	61
ACD_NVETO_04A(B)M	Veto Ch 4 - to AEM	62
ACD_NVETO_04A(B)P	Veto Ch 4 + to AEM	63
ACD_NVETO_03A(B)M	Veto Ch 3 - to AEM	64
ACD_NVETO_03A(B)P	Veto Ch 3 + to AEM	65
ACD_NVETO_02A(B)M	Veto Ch 2 - to AEM	66
ACD_NVETO_02A(B)P	Veto Ch 2 + to AEM	67
ACD_NVETO_01A(B)M	Veto Ch 1 - to AEM	68
ACD_NVETO_01A(B)P	Veto Ch 1 + to AEM	69
ACD_NVETO_00A(B)M	Veto Ch 0 - to AEM	70
ACD_NVETO_00A(B)P	Veto Ch 0 + to AEM	71
ACD_NSDATA_A(B)M	Data- to AEM	72
ACD_NSDATA_A(B)P	Data+ to AEM	73

ACD_NRST_A(B)M	Reset- to ACD	74
ACD_NRST_A(B)P	Reset+ to ACD	75
ACD_NSCMD_A(B)M	Command- to ACD	76
ACD_NSCMD_A(B)P	Command+ to ACD	77
ACD_CLK_A(B)M	Clock- to ACD	78
ACD_CLK_A(B)P	Clock+ to ACD	79
Spare Pins	Pins 8-16, 29, 35-39 not used	rest

Appendix 7: GARC Command Mnemonics and Functions

The following table represents each of the available GARC commands. Additionally, all GAFE commands are passed through the GARC. These command patterns are detailed in the document discussing the GAFE logic. There are two types of GARC commands – trigger commands (4 bits in length) and configuration commands (34 bits in length). The AEM-ACD ICD contains the authoritative formats for all command types, but they are repeated in the appendices of this document for convenience. The table below defines the command mnemonics that will be used in this test.

Note that a GAFE will process a write command either for the address hard-wired to the chip address pins or to an address of 'h1F, the GAFE broadcast address. A GAFE will process a read command only for an address identical to the hard-wired address. It is an operational constraint that, for any given ACD circuit board, each GAFE must have a unique address.

GARC Cmd No.	ACD Command Mnemonic	Rd/Wr Status	Select GARC=0 GAFE=1	Function Block	Register Number	No. of Data Bits	Command Function
1	Trigger_ZS	N/A	N/A	N/A	N/A	N/A	ACD Trigger, Zero-Suppression Enable
2	Trigger_NOZS	N/A	N/A	N/A	N/A	N/A	ACD Trigger, No Zero-Suppression
3	GARC_Reset	W	0	0	1	0	Generates reset for GARC and GAFE registers
4	Veto_Delay_Wr	W	0	0	2	5	Sets Delay from Disc_In to VETO Out
5	Veto_Delay_Rd	R	0	0	2	5	Reads contents of Veto_Delay register
6	GARC_Cal_Strobe	W	0	0	3	0	Sends Calibration Strobe signal to all GAFEs
7	HVBS_Level_Wr	W	0	0	8	12	Sets GARC register value from which HVBS may be commanded in the science mode
8	HVBS_Level_Rd	R	0	0	8	12	Reads contents of HVBS Level register
9	SAA_Level_Wr	W	0	0	9	12	Sets GARC register value from which HVBS may be commanded when in the SAA
10	SAA_Level_Rd	R	0	0	9	12	Reads contents of SAA Level register
11	Use_HV_Normal	W	0	0	10	0	Sends 12 bit value in HVBS Level register to the MAX5121 DAC
12	DAC_HVReg_Rd	R	0	0	10	0	Reads MAX5121 DAC Config Register
13	Use_SAA_Normal	W	0	0	11	0	Sends 12 bit value in SAA Level register to the MAX5121 DAC
14	DAC_SAAREg_Rd	R	0	0	11	0	Reads MAX5121 DAC Config Register
15	Hold_Delay_Wr	W	0	0	12	7	Sets GARC Hold Delay
16	Hold_Delay_Rd	R	0	0	12	7	Reads back value of GARC Hold Delay register
17	Veto_Width_Wr	W	0	0	13	3	Sets width of the VETO signals
18	Veto_Width_Rd	R	0	0	13	3	Reads back width of the VETO width register

19	HitMap_Width_Wr	W	0	0	14	4	Sets width of HitMap pulses
20	HitMap_Width_Rd	R	0	0	14	4	Reads back value in the HitMap width register
21	HitMap_Deathime_Wr	W	0	0	15	3	Sets stretch at end of HitMap pulses
22	HitMap_Deathime_Rd	R	0	0	15	3	Sets stretch at end of HitMap pulses
23	Look_At_Me	W	0	1	4	16	GARC interface selection command (primary/secondary)
24	HitMap_Delay_Wr	W	0	1	8	5	Sets delay from Disc_In to HitMap pulse
25	HitMap_Delay_Rd	R	0	1	8	5	Reads back contents of the HitMap_Delay register
26	PHA_EN0_Wr	W	0	1	9	16	Enables and Disables PHA readout enable, channels 0 -15
27	PHA_EN0_Rd	R	0	1	9	16	Reads back contents of the PHA_EN0 register
28	VETO_EN0_Wr	W	0	1	10	16	Enables and Disables VETO signals, channels 0 -15
29	VETO_EN0_Rd	R	0	1	10	16	Reads back contents of the VETO_EN0 register
30	PHA_EN1_Wr	W	0	1	12	2	Enables and Disables PHA readout enable, channels 16 & 17
31	PHA_EN1_Rd	R	0	1	12	2	Reads back contents of the PHA_EN1 register
32	VETO_EN1_Wr	W	0	1	13	2	Enables and Disables VETO signals, channels 16 & 17
33	VETO_EN1_Rd	R	0	1	13	2	Reads back contents of the VETP_EN1 register
34	Max_PHA_Wr	W	0	1	15	5	Sets the Maximum number (limit) of PHA to be returned in a single event data packet
35	Max_PHA_Rd	R	0	1	15	5	Reads back the contents of the Max_PHA register
36	GARC_Mode_Wr	W	0	2	8	11	Sets values of GARC mode bits
37	GARC_Mode_Rd	R	0	2	8	11	Reads back the value of the GARC mode register
38	GARC_Status	R	0	2	9	6	Reads back the value of the GARC status register
39	GARC_Cmd_Reg	R	0	2	10	16	Reads back the value of the GARC command register
40	GARC_Diagnostic	R	0	2	11	16	Reads back the value of the GARC diagnostic register
41	GARC_Cmd_Rejects	R	0	2	12	8	Reads back the number of rejected commands since reset
42	FREE_Board_ID	R	0	2	13	8	Reads back the FREE board serial number
43	GARC_Version	R	0	2	14	3	Reads back the GARC ASIC version number
44	PHA_Thresh00_Wr	W	0	3	8	12	Writes PHA ZS threshold for channel 00
45	PHA_Thresh00_Rd	R	0	3	8	12	Reads back PHA ZS threshold register for channel 00
46	PHA_Thresh01_Wr	W	0	3	9	12	Writes PHA ZS threshold for channel 01
47	PHA_Thresh01_Rd	R	0	3	9	12	Reads back PHA ZS threshold register for channel 01
48	PHA_Thresh02_Wr	W	0	3	10	12	Writes PHA ZS threshold for channel 02
49	PHA_Thresh02_Rd	R	0	3	10	12	Reads back PHA ZS threshold register for channel 02
50	PHA_Thresh03_Wr	W	0	3	11	12	Writes PHA ZS threshold for channel 03
51	PHA_Thresh03_Rd	R	0	3	11	12	Reads back PHA ZS threshold register for channel 03
52	PHA_Thresh04_Wr	W	0	3	12	12	Writes PHA ZS threshold for channel 04
53	PHA_Thresh04_Rd	R	0	3	12	12	Reads back PHA ZS threshold register for channel 04
54	PHA_Thresh05_Wr	W	0	3	13	12	Writes PHA ZS threshold for channel 05

55	PHA_Thresh05_Rd	R	0	3	13	12	Reads back PHA ZS threshold register for channel 05
56	PHA_Thresh06_Wr	W	0	3	14	12	Writes PHA ZS threshold for channel 06
57	PHA_Thresh06_Rd	R	0	3	14	12	Reads back PHA ZS threshold register for channel 06
58	PHA_Thresh07_Wr	W	0	4	8	12	Writes PHA ZS threshold for channel 07
59	PHA_Thresh07_Rd	R	0	4	8	12	Reads back PHA ZS threshold register for channel 07
60	PHA_Thresh08_Wr	W	0	4	9	12	Writes PHA ZS threshold for channel 08
61	PHA_Thresh08_Rd	R	0	4	9	12	Reads back PHA ZS threshold register for channel 08
62	PHA_Thresh09_Wr	W	0	4	10	12	Writes PHA ZS threshold for channel 09
63	PHA_Thresh09_Rd	R	0	4	10	12	Reads back PHA ZS threshold register for channel 09
64	PHA_Thresh10_Wr	W	0	4	11	12	Writes PHA ZS threshold for channel 10
65	PHA_Thresh10_Rd	R	0	4	11	12	Reads back PHA ZS threshold register for channel 10
66	PHA_Thresh11_Wr	W	0	4	12	12	Writes PHA ZS threshold for channel 11
67	PHA_Thresh11_Rd	R	0	4	12	12	Reads back PHA ZS threshold register for channel 11
68	PHA_Thresh12_Wr	W	0	4	13	12	Writes PHA ZS threshold for channel 12
69	PHA_Thresh12_Rd	R	0	4	13	12	Reads back PHA ZS threshold register for channel 12
70	PHA_Thresh13_Wr	W	0	4	14	12	Writes PHA ZS threshold for channel 13
71	PHA_Thresh13_Rd	R	0	4	14	12	Reads back PHA ZS threshold register for channel 13
72	PHA_Thresh14_Wr	W	0	5	8	12	Writes PHA ZS threshold for channel 14
73	PHA_Thresh14_Rd	R	0	5	8	12	Reads back PHA ZS threshold register for channel 14
74	PHA_Thresh15_Wr	W	0	5	9	12	Writes PHA ZS threshold for channel 15
75	PHA_Thresh15_Rd	R	0	5	9	12	Reads back PHA ZS threshold register for channel 15
76	PHA_Thresh16_Wr	W	0	5	10	12	Writes PHA ZS threshold for channel 16
77	PHA_Thresh16_Rd	R	0	5	10	12	Reads back PHA ZS threshold register for channel 16
78	PHA_Thresh17_Wr	W	0	5	11	12	Writes PHA ZS threshold for channel 17
79	PHA_Thresh17_Rd	R	0	5	11	12	Reads back PHA ZS threshold register for channel 17
80	ADC_TACQ_Wr	W	0	5	12	6	Sets ADC Acquisition Time from Hold to Start of Conversion
81	ADC_TACQ_Rd	R	0	5	12	6	Reads back contents of the ADC_TACQ register
82	GAFE_Mode_Wr	W	1	GAFE Addr	0	16	Writes the GAFE mode register for the ASIC addressed
83	GAFE_Mode_Rd	R	1	GAFE Addr	0	16	Reads the GAFE mode register contents for the ASIC addressed
84	GAFE_VETO_Wr	W	1	GAFE Addr	1	6	Writes the DAC1 register in the GAFE addressed
85	GAFE_VETO_Rd	R	1	GAFE Addr	1	6	Reads back the contents of the DAC1 register in the addressed GAFE
86	GAFE_VERNIER_Wr	W	1	GAFE Addr	2	6	Writes the DAC2 register in the GAFE addressed
87	GAFE_VERNIER_Rd	R	1	GAFE Addr	2	6	Reads back the contents of the DAC2 register in the addressed GAFE
88	GAFE_HLD_Wr	W	1	GAFE Addr	3	6	Writes the DAC3 register in the GAFE addressed
89	GAFE_HLD_Rd	R	1	GAFE Addr	3	6	Reads back the contents of the DAC3 register in the addressed GAFE
90	GAFE_BIAS_Wr	W	1	GAFE Addr	4	6	Writes the DAC4 register in the GAFE addressed
91	GAFE_BIAS_Rd	R	1	GAFE Addr	4	6	Reads back the contents of the DAC4 register in the addressed GAFE
92	GAFE_TCI_Wr	W	1	GAFE	5	6	Writes the DAC5 register in the GAFE

				Addr			addressed
93	GAFE_TCI_Rd	R	1	GAFE Addr	5	6	Reads back the contents of the DAC5 register in the addressed GAFE
94	GAFE_Version	R	1	GAFE Addr	6	3	Reads back the GAFE ASIC version
95	GAFE_Write_Ctr	R	1	GAFE Addr	7	8	Reads back the contents of the GAFE write counter register
96	GAFE_Reject_Ctr	R	1	GAFE Addr	8	8	Reads back the contents of the GAFE command reject register
97	GAFE_Cmd_Ctr	R	1	GAFE Addr	9	8	Reads back the contents of the GAFE command counter
98	GAFE_Chip_Addr	R	1	GAFE Addr	10	5	Reads back the hardwired address of a GAFE ASIC

Appendix 8: Test Results Record

All measurements and test results will be recorded in this Test Results Record. A copy of the entire this entire document is not required each time the test is performed. The appendix will serve as the official record each time this test is performed.

3.0 QA Signoff

Quality Assurance approval to proceed with the test - _____

(NOTE: If QA gives verbal approval to proceed but is not in attendance for the test, record the date and the time of approval.)

3.1 FREE Assembly Identification

The test conductor for this test is: _____

Date of the Test: _____

The serial number of the FREE card assembly is: _____

The identification/Serial Number listed on the GARC ASIC is: _____

The identification/Serial Numbers listed on the GAFE ASICs are:

GAFE Channel	ASIC Serial Number
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	

3.2 Test Equipment Utilized

Instrument Type	Manufacturer & Model Number	NASA ID Number	Calibration Due Date
Power Supply for HVBS +28V			
Power Supply for FREE & GSE +3.3V & 5.0V			
Multimeter 1			
Multimeter 2			
Test Pulse Generator			
Oscilloscope			
Pulse Generator			
Charge Terminator Box			
GSE Used			

4.1 Verification of Proper LVDS Terminations

Test Omitted - _____

Measurement No.	Signal Pair	Pin 1	Pin 2	Expected Resistance	Verified
1	ACD_NVETO_16_A	JP1 - 17	JP1 - 18	$95\ \Omega < R < 105\ \Omega$	
2	ACD_NVETO_17_A	JP1 - 19	JP1 - 20	$95\ \Omega < R < 105\ \Omega$	
3	ACD_CNO_A	JP1 - 21	JP1 - 22	$95\ \Omega < R < 105\ \Omega$	
4	ACD_NVETO_15_A	JP1 - 40	JP1 - 41	$95\ \Omega < R < 105\ \Omega$	
5	ACD_NVETO_14_A	JP1 - 42	JP1 - 43	$95\ \Omega < R < 105\ \Omega$	
6	ACD_NVETO_13_A	JP1 - 44	JP1 - 45	$95\ \Omega < R < 105\ \Omega$	
7	ACD_NVETO_12_A	JP1 - 46	JP1 - 47	$95\ \Omega < R < 105\ \Omega$	
8	ACD_NVETO_11_A	JP1 - 48	JP1 - 49	$95\ \Omega < R < 105\ \Omega$	
9	ACD_NVETO_10_A	JP1 - 50	JP1 - 51	$95\ \Omega < R < 105\ \Omega$	
10	ACD_NVETO_09_A	JP1 - 52	JP1 - 53	$95\ \Omega < R < 105\ \Omega$	
11	ACD_NVETO_08_A	JP1 - 54	JP1 - 55	$95\ \Omega < R < 105\ \Omega$	

12	ACD_NVETO_07_A	JP1 - 56	JP1 - 57	$95\ \Omega < R < 105\ \Omega$	
13	ACD_NVETO_06_A	JP1 - 58	JP1 - 59	$95\ \Omega < R < 105\ \Omega$	
14	ACD_NVETO_05_A	JP1 - 60	JP1 - 61	$95\ \Omega < R < 105\ \Omega$	
15	ACD_NVETO_04_A	JP1 - 62	JP1 - 63	$95\ \Omega < R < 105\ \Omega$	
16	ACD_NVETO_03_A	JP1 - 64	JP1 - 65	$95\ \Omega < R < 105\ \Omega$	
17	ACD_NVETO_02_A	JP1 - 66	JP1 - 67	$95\ \Omega < R < 105\ \Omega$	
18	ACD_NVETO_01_A	JP1 - 68	JP1 - 69	$95\ \Omega < R < 105\ \Omega$	
19	ACD_NVETO_00_A	JP1 - 70	JP1 - 71	$95\ \Omega < R < 105\ \Omega$	
20	ACD_NSDATA_A	JP1 - 72	JP1 - 73	$95\ \Omega < R < 105\ \Omega$	
21	ACD_NNST_A	JP1 - 74	JP1 - 75	$R > 100\ k\Omega$	
22	ACD_NSCMD_A	JP1 - 76	JP1 - 77	$R > 100\ k\Omega$	
23	ACD_CLK_A	JP1 - 78	JP1 - 79	$R > 100\ k\Omega$	
24	ACD_NVETO_16_B	JS2 - 17	JS2 - 18	$95\ \Omega < R < 105\ \Omega$	
25	ACD_NVETO_17_B	JS2 - 19	JS2 - 20	$95\ \Omega < R < 105\ \Omega$	
26	ACD_CNO_B	JS2 - 21	JS2 - 22	$95\ \Omega < R < 105\ \Omega$	
27	ACD_NVETO_15_B	JS2 - 40	JS2 - 41	$95\ \Omega < R < 105\ \Omega$	
28	ACD_NVETO_14_B	JS2 - 42	JS2 - 43	$95\ \Omega < R < 105\ \Omega$	
29	ACD_NVETO_13_B	JS2 - 44	JS2 - 45	$95\ \Omega < R < 105\ \Omega$	
30	ACD_NVETO_12_B	JS2 - 46	JS2 - 47	$95\ \Omega < R < 105\ \Omega$	
31	ACD_NVETO_11_B	JS2 - 48	JS2 - 49	$95\ \Omega < R < 105\ \Omega$	
32	ACD_NVETO_10_B	JS2 - 50	JS2 - 51	$95\ \Omega < R < 105\ \Omega$	
33	ACD_NVETO_09_B	JS2 - 52	JS2 - 53	$95\ \Omega < R < 105\ \Omega$	
34	ACD_NVETO_08_B	JS2 - 54	JS2 - 55	$95\ \Omega < R < 105\ \Omega$	
35	ACD_NVETO_07_B	JS2 - 56	JS2 - 57	$95\ \Omega < R < 105\ \Omega$	
36	ACD_NVETO_06_B	JS2 - 58	JS2 - 59	$95\ \Omega < R < 105\ \Omega$	
37	ACD_NVETO_05_B	JS2 - 60	JS2 - 61	$95\ \Omega < R < 105\ \Omega$	
38	ACD_NVETO_04_B	JS2 - 62	JS2 - 63	$95\ \Omega < R < 105\ \Omega$	
39	ACD_NVETO_03_B	JS2 - 64	JS2 - 65	$95\ \Omega < R < 105\ \Omega$	
40	ACD_NVETO_02_B	JS2 - 66	JS2 - 67	$95\ \Omega < R < 105\ \Omega$	
41	ACD_NVETO_01_B	JS2 - 68	JS2 - 69	$95\ \Omega < R < 105\ \Omega$	
42	ACD_NVETO_00_B	JS2 - 70	JS2 - 71	$95\ \Omega < R < 105\ \Omega$	
43	ACD_NSDATA_B	JS2 - 72	JS2 - 73	$95\ \Omega < R < 105\ \Omega$	
44	ACD_NNST_B	JS2 - 74	JS2 - 75	$R > 100\ k\Omega$	
45	ACD_NSCMD_B	JS2 - 76	JS2 - 77	$R > 100\ k\Omega$	
46	ACD_CLK_B	JS2 - 78	JS2 - 79	$R > 100\ k\Omega$	

4.2 Stray Voltage Test at the AEM Interface

Test Omitted - _____

Meas. No.	AEM Interface Pin	Signal Name	Expected Voltage	Measured Voltage
1	JP1 - 1	ACD_VDD_0A	+3.3V	
2	JP1 - 3	ACD_VDD_1A	+3.3V	
3	JP1 - 4	ACD_VDD_2A	+3.3V	
4	JP1 - 5	ACD_28V_0A	+28V	
5	JP1 - 7	ACD_28V_1A	+28V	
6	JP1 - 17	ACD_NVETO_16AP	$0 < V < 3.3$	
7	JP1 - 18	ACD_NVETO_16AM	$0 < V < 3.3$	
8	JP1 - 19	ACD_NVETO_17AP	$0 < V < 3.3$	
9	JP1 - 20	ACD_NVETO_17AM	$0 < V < 3.3$	

10	JP1 - 21	ACD_NCNO_AP	0< V < 3.3	
11	JP1 - 22	ACD_NCNO_AM	0< V < 3.3	
12	JP1 - 23	ACD_HV_AP	0< V < 3.3	
13	JP1 - 24	ACD_HV_AM	0< V < 3.3	
14	JP1 - 25	ACD_TEMP_AP	TBD	
15	JP1 - 26	ACD_TEMP_AM	TBD	
16	JP1 - 30	ACD_GND_0A	0	
17	JP1 - 31	ACD_GND_1A	0	
18	JP1 - 32	ACD_GND_2A	0	
19	JP1 - 33	ACD_28V_RTN_0A	0	
20	JP1 - 34	ACD_28V_RTN_1A	0	
21	JP1 - 40	ACD_NVETO_15AM	V < 200mV	
22	JP1 - 41	ACD_NVETO_15AP	V < 200mV	
23	JP1 - 42	ACD_NVETO_14AM	V < 200mV	
24	JP1 - 43	ACD_NVETO_14AP	V < 200mV	
25	JP1 - 44	ACD_NVETO_13AM	V < 200mV	
26	JP1 - 45	ACD_NVETO_13AP	V < 200mV	
27	JP1 - 46	ACD_NVETO_12AM	V < 200mV	
28	JP1 - 47	ACD_NVETO_12AP	V < 200mV	
29	JP1 - 48	ACD_NVETO_11AM	V < 200mV	
31	JP1 - 50	ACD_NVETO_10AM	V < 200mV	
32	JP1 - 51	ACD_NVETO_10AP	V < 200mV	
33	JP1 - 52	ACD_NVETO_09AM	V < 200mV	
34	JP1 - 53	ACD_NVETO_09AP	V < 200mV	
35	JP1 - 54	ACD_NVETO_08AM	V < 200mV	
36	JP1 - 55	ACD_NVETO_08AP	V < 200mV	
37	JP1 - 56	ACD_NVETO_07AM	V < 200mV	
38	JP1 - 57	ACD_NVETO_07AP	V < 200mV	
39	JP1 - 58	ACD_NVETO_06AM	V < 200mV	
40	JP1 - 59	ACD_NVETO_06AP	V < 200mV	
41	JP1 - 60	ACD_NVETO_05AM	V < 200mV	
42	JP1 - 61	ACD_NVETO_05AP	V < 200mV	
43	JP1 - 62	ACD_NVETO_04AM	V < 200mV	
44	JP1 - 63	ACD_NVETO_04AP	V < 200mV	
45	JP1 - 64	ACD_NVETO_03AM	V < 200mV	
46	JP1 - 65	ACD_NVETO_03AP	V < 200mV	
47	JP1 - 66	ACD_NVETO_02AM	V < 200mV	
48	JP1 - 67	ACD_NVETO_02AP	V < 200mV	
49	JP1 - 68	ACD_NVETO_01AM	V < 200mV	
50	JP1 - 69	ACD_NVETO_01AP	V < 200mV	
51	JP1 - 70	ACD_NVETO_00AM	V < 200mV	
52	JP1 - 71	ACD_NVETO_00AP	V < 200mV	
53	JP1 - 72	ACD_NSDATA_AM	V < 200mV	
54	JP1 - 73	ACD_NSDATA_AP	V < 200mV	
55	JP1 - 74	ACD_NRST_AM	V < 200mV	
56	JP1 - 75	ACD_NRST_AP	V > 2.3V	
57	JP1 - 76	ACD_NSCMD_AM	V < 200mV	
58	JP1 - 77	ACD_NSCMD_AP	V > 2.3V	
59	JP1 - 78	ACD_CLK_AM	1.1V< V < 1.5V	
60	JP1 - 79	ACD_CLK_AP	1.1V< V < 1.5V	
61	JS2 - 1	ACD_VDD_0B	+3.3V	

62	JS2 - 3	ACD_VDD_1B	+3.3V	
63	JS2 - 4	ACD_VDD_2B	+3.3V	
64	JS2 - 5	ACD_28V_0B	+28V	
65	JS2 - 7	ACD_28V_1B	+28V	
66	JS2 - 17	ACD_NVETO_16BP	V < 200mV	
67	JS2 - 18	ACD_NVETO_16BM	V < 200mV	
68	JS2 - 19	ACD_NVETO_17BP	V < 200mV	
69	JS2 - 20	ACD_NVETO_17BM	V < 200mV	
70	JS2 - 21	ACD_NCNO_BP	V < 200mV	
71	JS2 - 22	ACD_NCNO_BM	V < 200mV	
72	JS2 - 23	ACD_HV_BP	V < 200mV	
73	JS2 - 24	ACD_HV_BM	V < 200mV	
74	JS2 - 25	ACD_TEMP_BP	TBD	
75	JS2 - 26	ACD_TEMP_BM	TBD	
76	JS2 - 30	ACD_GND_0B	0	
77	JS2 - 31	ACD_GND_1B	0	
78	JS2 - 32	ACD_GND_2B	0	
79	JS2 - 33	ACD_28V_RTN_0B	0	
80	JS2 - 34	ACD_28V_RTN_1B	0	
81	JS2 - 40	ACD_NVETO_15BM	V < 200mV	
82	JS2 - 41	ACD_NVETO_15BP	V < 200mV	
83	JS2 - 42	ACD_NVETO_14BM	V < 200mV	
84	JS2 - 43	ACD_NVETO_14BP	V < 200mV	
85	JS2 - 44	ACD_NVETO_13BM	V < 200mV	
86	JS2 - 45	ACD_NVETO_13BP	V < 200mV	
87	JS2 - 46	ACD_NVETO_12BM	V < 200mV	
88	JS2 - 47	ACD_NVETO_12BP	V < 200mV	
89	JS2 - 48	ACD_NVETO_11BM	V < 200mV	
90	JS2 - 49	ACD_NVETO_11BP	V < 200mV	
91	JS2 - 50	ACD_NVETO_10BM	V < 200mV	
92	JS2 - 51	ACD_NVETO_10BP	V < 200mV	
93	JS2 - 52	ACD_NVETO_09BM	V < 200mV	
94	JS2 - 53	ACD_NVETO_09BP	V < 200mV	
95	JS2 - 54	ACD_NVETO_08BM	V < 200mV	
96	JS2 - 55	ACD_NVETO_08BP	V < 200mV	
97	JS2 - 56	ACD_NVETO_07BM	V < 200mV	
98	JS2 - 57	ACD_NVETO_07BP	V < 200mV	
99	JS2 - 58	ACD_NVETO_06BM	V < 200mV	
100	JS2 - 59	ACD_NVETO_06BP	V < 200mV	
101	JS2 - 60	ACD_NVETO_05BM	V < 200mV	
102	JS2 - 61	ACD_NVETO_05BP	V < 200mV	
103	JS2 - 62	ACD_NVETO_04BM	V < 200mV	
104	JS2 - 63	ACD_NVETO_04BP	V < 200mV	
105	JS2 - 64	ACD_NVETO_03BM	V < 200mV	
106	JS2 - 65	ACD_NVETO_03BP	V < 200mV	
107	JS2 - 66	ACD_NVETO_02BM	V < 200mV	
108	JS2 - 67	ACD_NVETO_02BP	V < 200mV	
109	JS2 - 68	ACD_NVETO_01BM	V < 200mV	
110	JS2 - 69	ACD_NVETO_01BP	V < 200mV	
111	JS2 - 70	ACD_NVETO_00BM	V < 200mV	
112	JS2 - 71	ACD_NVETO_00BP	V < 200mV	

113	JS2 - 72	ACD_NSDATA_BM	$V < 200\text{mV}$	
114	JS2 - 73	ACD_NSDATA_BP	$V < 200\text{mV}$	
115	JS2 - 74	ACD_NRST_BM	$V < 200\text{mV}$	
116	JS2 - 75	ACD_NRST_BP	$V > 2.3\text{V}$	
117	JS2 - 76	ACD_NSCMD_BM	$V < 200\text{mV}$	
118	JS2 - 77	ACD_NSCMD_BP	$V > 2.3\text{V}$	
119	JS2 - 78	ACD_CLK_BM	$1.1\text{V} < V < 1.5\text{V}$	
120	JS2 - 79	ACD_CLK_BP	$1.1\text{V} < V < 1.5\text{V}$	

5.1 GARC and GAFE Registers Initial Reset Test

+3.3V Current - _____

+28V Current - _____

Turn On reset verified - _____

FREE Board ID - _____

Reset Command verified - _____

5.2 FREE Power Measurement at the Nominal Power Supply Voltage

GARC Mode	GARC_Mode_Wr Data Argument	+3.3V Current Measured (mA)	+3.3V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		$200 \pm 10\text{mA}$
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		$150 \pm 10\text{mA}$
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		$150 \pm 10\text{mA}$
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		$95 \pm 10\text{mA}$

5.3 FREE Power Measurement at the Minimum Power Supply Voltage

GARC Mode	GARC_Mode_Wr Data Argument	+3.0V Current Measured (mA)	+3.0V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		$170 \pm 10\text{mA}$
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		$125 \pm 10\text{mA}$
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		$125 \pm 10\text{mA}$
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		$75 \pm 10\text{mA}$

5.4 FREE Power Measurement at the Maximum Power Supply Voltage

GARC Mode	GARC_Mode_Wr Data Argument	+3.6V Current Measured (mA)	+3.6V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		230 ± 10mA
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		175 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		175 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		115 ± 10mA

5.5 GARC/GAFE Register Read/Write Tests

This test may be automated using the LabView GSE via the **Register Test VI**.

Test Completed - _____

5.45 Initial GAFE Logic Reset Test

This test may be automated using the LabView GSE via the **GAFE Logic Reset Test.txt** script.

Test Completed - _____

5.46 GAFE ASIC Broadcast Command Functional Verification

This test may be automated using the LabView GSE via the **GAFE Broadcast Command Test.txt** script.

Test Completed - _____

5.47 Test of the GARC Parity Test

This test may be automated using the LabView GSE via the **GARC Parity Test.txt** script.

Test Completed - _____

5.48 Test of the Look-At-Me Circuitry

This test may be automated using the LabView GSE via the **GARC Look At Me Test.txt** script.

Test Completed - _____

5.49 Maximum PHA Return Test

This test may be automated using the LabView GSE via the **GARC Max PHA Return Test.tst** script.

Test Completed - _____

5.50 PHA Enable/Disable Test

This test may be automated using the LabView GSE via the **GARC PHA Enable Test.tst** script.

Test Completed - _____

5.51 PHA Threshold Verification Test

This test may be automated using the LabView GSE via the **GARC PHA Threshold Test.txt** script.

Test Completed - _____

5.52 HVBS Connector Checks and Circuit Check-Outs

This test may be automated using the LabView GSE via the **GARC HVBS DAC Level Test.txt** script.

Meas. No.	DAC SETTING	JHV1-4 (HV-DACP)	JHV1-9 (HV-DACN)
1	0		
2	200		
3	400		
4	600		
5	800		
6	1000		
7	1200		
8	1400		
9	1600		
10	1800		
11	2000		
12	2200		
13	2400		
14	2600		
15	2800		
16	3000		
17	3200		
18	3400		
19	3600		
20	3800		
21	4000		
22	4095		

Meas. No.	DAC SETTING	JHV2-4 (HV-DACP)	JHV2-9 (HV-DACN)
1	0		
2	4095		

5.53 Test of the HVBS Triple Modular Redundancy Circuitry

This test may be automated using the LabView GSE via the GARC HV Enable Test.tst script.

Voltage at JHV1-5 when Enabled - _____ (3.7V)

Voltage at JHV2-5 when Enabled - _____ (3.7V)

Voltage at JHV1-5 when Disabled - _____ (0.04V)

Voltage at JHV2-5 when Disabled - _____ (0.04V)

Test Completed - _____

5.54 TEST OF THE SAA/HV NORMAL MODES

Set HVBS Normal to 2048 and SAA to 1024

3. Using a multimeter at the JHV1 pin 4 connector, verify the HV_DACP voltage is 1.4V

Value observed: _____

4. Send the Use_HV_Nominal command. Verify that the HV_DACP voltage is 2.1V.

Value observed: _____

5. Send the DAC_HVReg_Rd command and verify the return data pattern from the MAX5121 DAC is decimal 10240.

Value observed: _____

6. Send the Use_SAA_Level command. Verify that the HV_DACP voltage is 1.8V.

Value observed: _____

7. Send the DAC_SAAReg_Rd command and verify the return data pattern from the MAX5121 DAC is decimal 9216.

Value observed: _____

8. Send the GARC_Reset command. Verify that the HV_DACP voltage is 1.4V..

Value observed: _____

5.55 Test of the GARC LVDS Circuitry Driver Currents

Test Omitted - _____

Obtain a scope picture of the nominal pulse Amplitude Swings

GARC Signal	Signal Location	+ DC Level (V) Enabled	- DC Level (V) Enabled	+ DC Level (V) Disabled	- DC Level (V) Disabled
ACD NVETO 12B	R1				
ACD NVETO 12A	R2				
ACD NVETO 11B	R3				
ACD NVETO 11A	R4				
ACD NVETO 10B	R5				
ACD NVETO 10A	R6				
ACD NVETO 09B	R7				
ACD NVETO 09A	R8				
ACD NVETO 08B	R9				
ACD NVETO 08A	R10				
ACD NVETO 07B	R11				
ACD NVETO 07A	R12				
ACD NVETO 06B	R13				
ACD NVETO 06A	R14				
ACD NVETO 05B	R15				
ACD NVETO 05A	R16				
ACD NVETO 04B	R17				
ACD NVETO 04A	R18				
ACD NVETO 03B	R19				
ACD NVETO 03A	R20				
ACD NVETO 02B	R21				
ACD NVETO 02A	R22				
ACD NVETO 01B	R23				
ACD NVETO 01A	R24				
ACD NVETO 00B	R25				
ACD NVETO 00A	R26				
ACD CNO B	R27				
ACD CNO A	R28				
ACD NVETO 13B	R31				
ACD NVETO 13A	R32				
ACD NVETO 14B	R33				
ACD NVETO 14A	R34				
ACD NVETO 15B	R35				
ACD NVETO 15A	R36				
ACD NVETO 16B	R37				
ACD NVETO 16A	R38				
ACD NVETO 17B	R39				
ACD NVETO 17A	R40				
ACD NSDATA B	R29				
ACD NSDATA A	R30				

5.56 FREE Circuit Assembly Characterization Tests

Test Omitted - _____

Characterization of each of the 18 GAFE chips

Run script _____

Data file used: _____

All printouts are to be attached to the Test Results Record.

Settings to be used based on the characterization test:

Create a configuration script for the FREE board under test.

File Name _____

GAFE Configuration

GAFE Channel	Mode	VETO DAC	VETO VERNIER	HLD DAC	BIAS DAC	TCI DAC
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						

GARC Configuration

Register	Test Value
Veto_Delay	
HVBS_Level	
SAA_Level	
Hold_Delay	
Veto_Width	

HitMap_Width	
HitMap_Deadtime	
HitMap_Delay	
PHA_En0	
PHA_En1	
VETO_En0	
VETO_En1	
Max_PHA	
GARC_Mode	
FREE_Board_ID	
GARC_Version	
PHA_Threshold_00	
PHA_Threshold_01	
PHA_Threshold_02	
PHA_Threshold_03	
PHA_Threshold_04	
PHA_Threshold_05	
PHA_Threshold_06	
PHA_Threshold_07	
PHA_Threshold_08	
PHA_Threshold_09	
PHA_Threshold_10	
PHA_Threshold_11	
PHA_Threshold_12	
PHA_Threshold_13	
PHA_Threshold_14	
PHA_Threshold_15	
PHA_Threshold_16	
PHA_Threshold_17	
ADC_TACQ	

5.57 TCI and GAFE ADC Test

All printouts are to be attached to the Test Results Record.

Data File Name - _____

Test Complete - _____

5.58 Power Supply Rail Tests – 3.6V to 3.0V

Test Omitted - _____

+3.6V current - _____

Register Test Complete - _____

DAC Checkout. This test may be automated using the LabView GSE via the **GARC HVBS DAC Level Test.txt** script.

Meas. No.	DAC SETTING	JHV1-4 to JHV1-9 (HV-DAC)
1	0	
2	1000	
3	2000	
4	3000	
5	4000	
6	4095	

Characterization/TCI Test File Name - _____

All printouts are to be attached to the Test Results Record.

+3.0V current - _____

Register Test Complete - _____

DAC Checkout. This test may be automated using the LabView GSE via the **GARC HVBS DAC Level Test.txt** script.

Meas. No.	DAC SETTING	JHV1-4 to JHV1-9 (HV-DAC)
1	0	
2	1000	
3	2000	
4	3000	
5	4000	
6	4095	

Characterization/TCI Test File Name - _____

All printouts are to be attached to the Test Results Record.

END OF SHORT FUNCTIONAL TEST